LẬP TRÌNH C TRÊN WIN

Lương Văn Vân - Khoa CNTT

Programming Language

Ngôn ngữ lập trình?

Khái niệm

Mục đích

Ngôn ngữ lập trình?

- Ngôn ngữ lập trình là hệ thống hữu hạn các ký hiệu, quy ước về ngữ pháp dùng để xây dựng các chương trình.
- Hướng dẫn máy tính giải một bài toán hay một yêu cầu đặt ra.

MỘT SỐ TIÊU ĐIỂM CỦA MÔN HỌC

- **Tổng số tiết: 90 tiết**
  - Lý thuyết: 25 tiết
  - Bài tập: 5 tiết
  - Thực hành: 60 tiết
- **Sinh viên cần phải được học trước các môn**
  - Kỹ thuật lập trình cơ bản.
  - Lập trình C.
  - Kỹ thuật lập trình hướng đối tượng.
- **Thi kết thúc môn học bằng hình thức thi thực hành.**

NỘI DUNG

C1. Môi trường lập trình trong Windows

C2. Thực đơn, thanh công cụ và thanh trạng thái

C3. Đồ họa và xử lý các thông điệp đầu vào

C4. Hộp hội thoại và các điều khiển

TÀI LIỆU THAM KHẢO

"Bài giảng lập trình C trên Windows"
Lương Văn Vân - Khoa CNTT

"Lập trình C trên Windows"
Đặng Văn Đức, NXB Khoa học kỹ thuật

"Lập trình Windows bằng Visual C++"
Đặng Văn Đức - Lê Quốc Hưng, Giáo Dục

"http://www.codeproject.com"
"http://www.codeguru.com"

MFC

Mọi thắc mắc xin liên hệ

Di động

Email

Nơi làm việc

0983.890088

luongvandlu@gmail

Văn phòng khoa

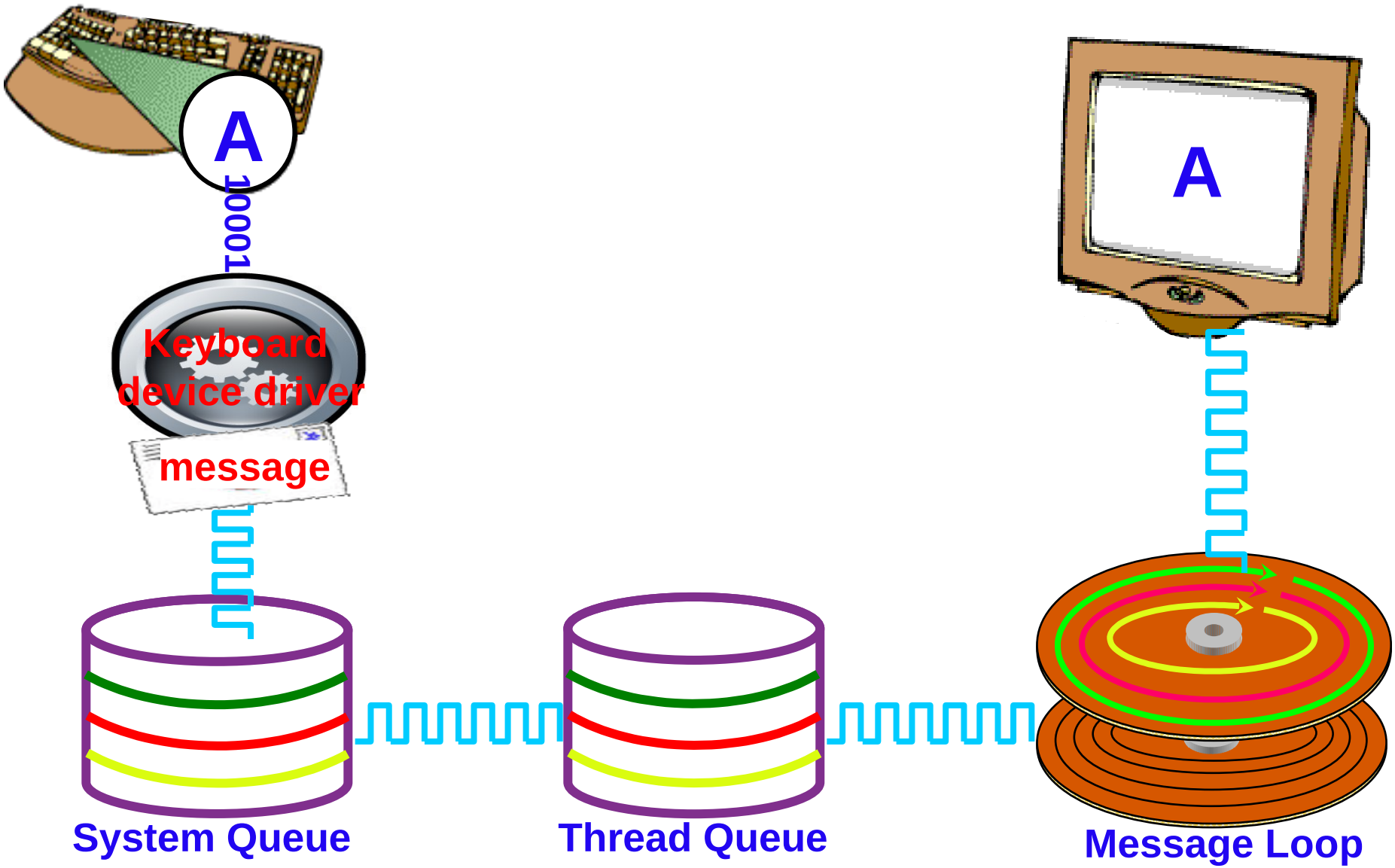
Môi trường lập trình trong Windows

- Đặc điểm của môi trường windows
- Vòng lặp thông điệp
- Giới thiệu MFC
- Tạo chương trình bằng AppWizard
- Phân tích các tập tin của một ứng dụng

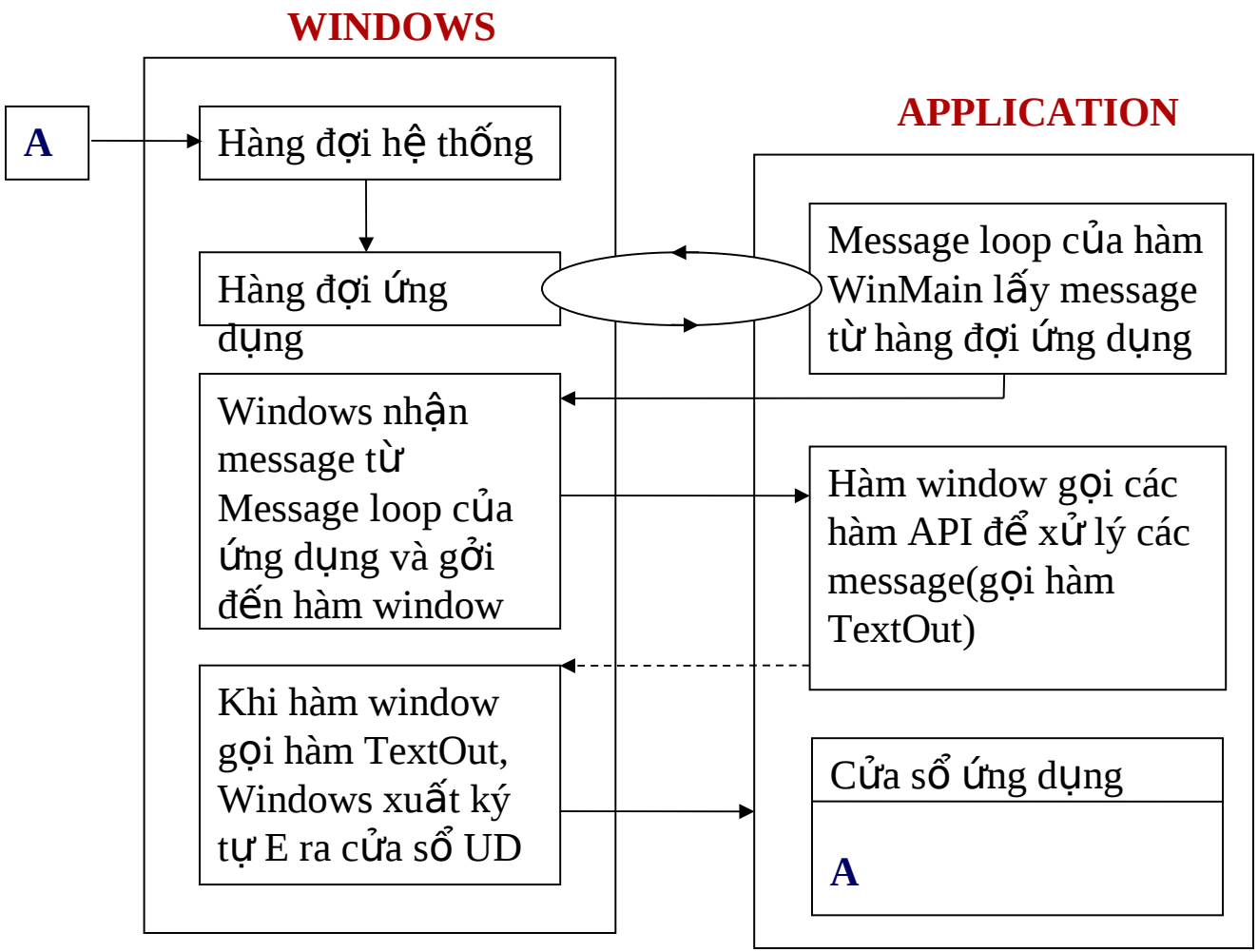
Đặc điểm của môi trường windows

1. Giao diện người dùng kiểu đồ họa (GUI)
2. Đa nhiệm
3. Quản lý bộ nhớ
4. Tư tưởng hướng đối tượng
5. Giao diện đồ họa độc lập với thiết bị
6. Kiến trúc hướng thông điệp
7. Thủ tục cửa sổ
8. Tài nguyên

Vòng lặp thông điệp



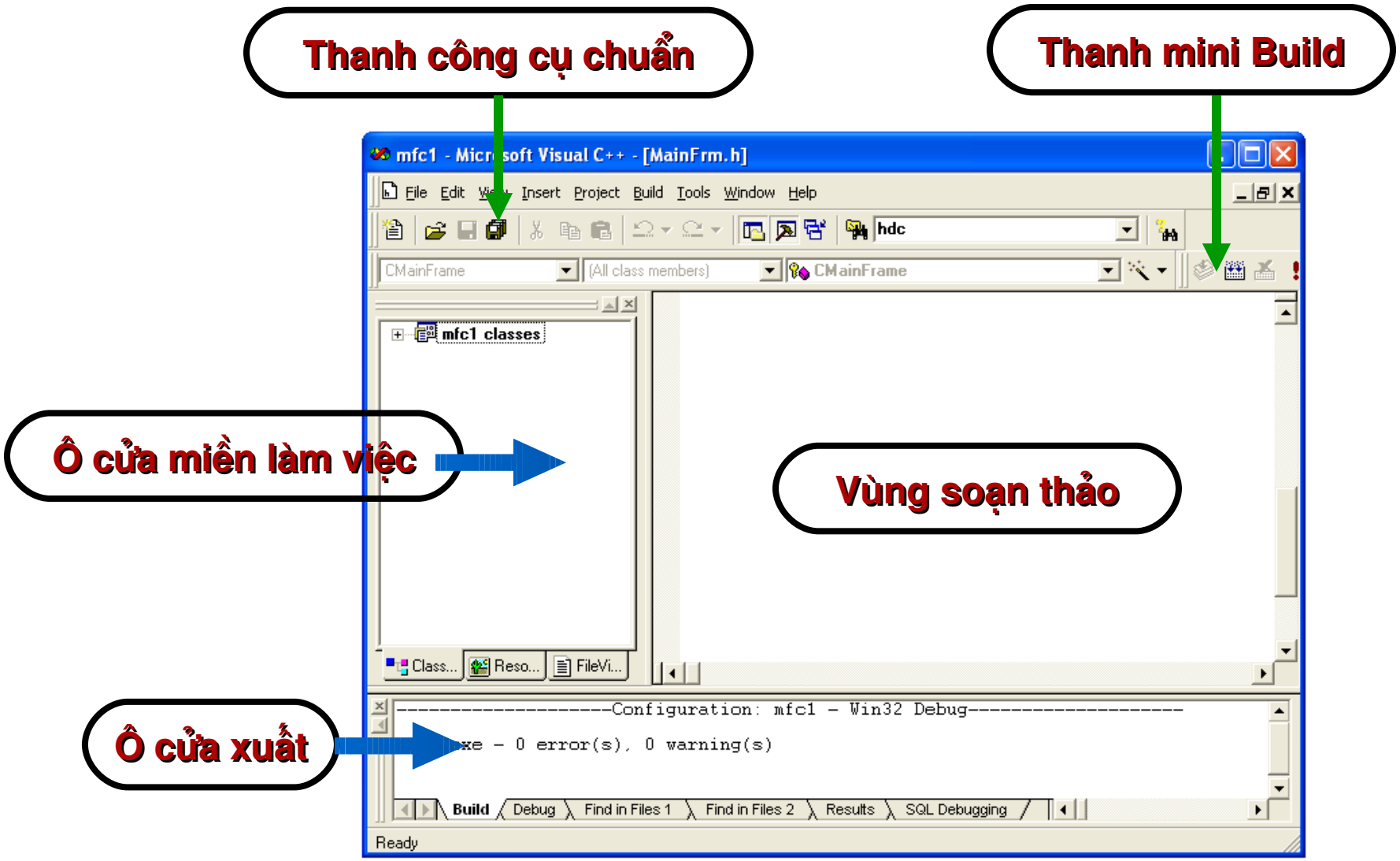
Cơ chế hoạt động của thông điệp



Giới thiệu MFC

- **Đặc điểm lập trình MFC**
  - **Microsoft Foundation Class**: Tập hợp các lớp định nghĩa sẵn.
  - Biểu diễn cách tiếp cận hướng đối tượng đến lập trình **Windows** và gói các **Windows API.**
  - Cho phép **Lập trình viên** ít phải lo lắng về giao diện **Windows.**
  - Làm đơn giản tiến trình phát triển mã trình cho các loại máy có hệ điều hành khác nhau.
  - Có hơn 130 lớp.
  - ……..

Môi trường phát triển Visual C++

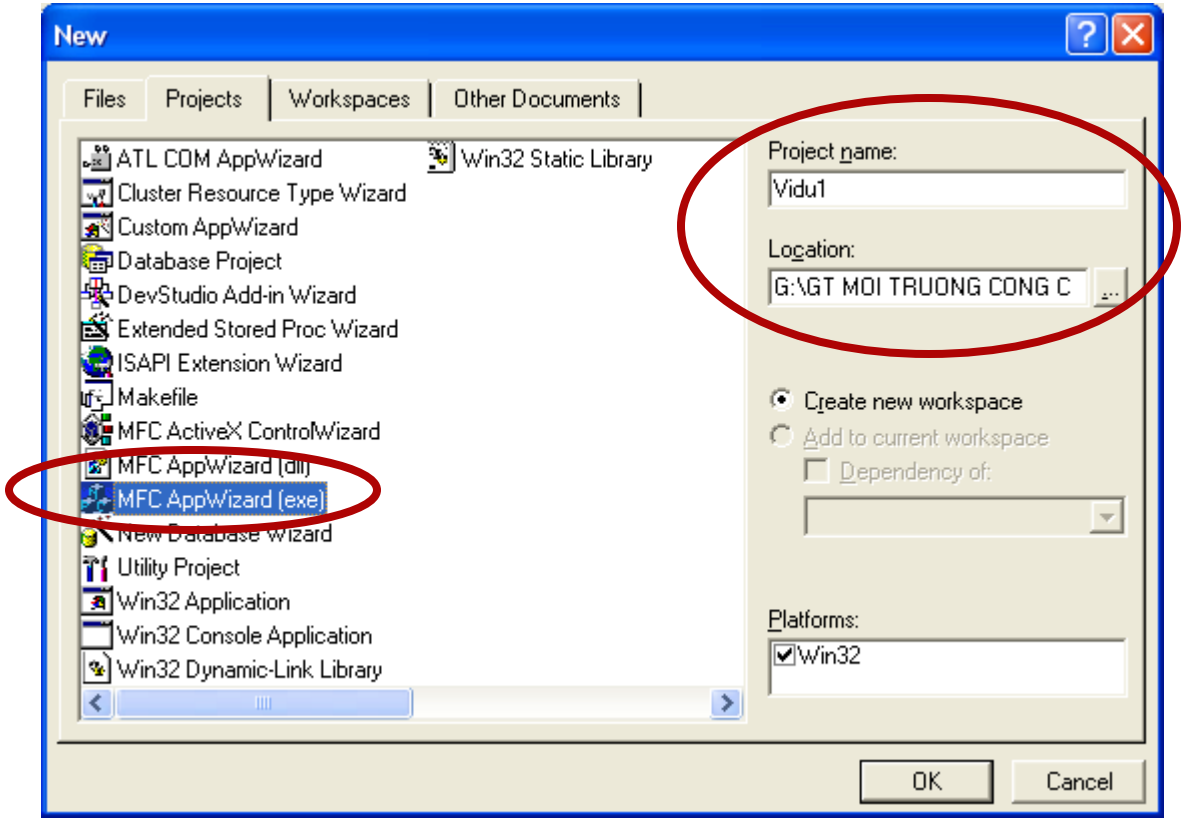


Môi trường phát triển Visual C++

- **Miền làm việc:** chứa các thành phần của ứng dụng:
  - Class View: điều hành và thao tác mã nguồn trên mức lớp.
  - Resource View: tìm và chọn lọc tài nguyên của ứng dụng.
  - File View: xem và điều hành tất cả các file.
- **Ô cửa sổ:** thông báo lỗi và lời cảnh báo của trình biên dịch.
- **Vùng soạn thảo:** nơi các cửa sổ soạn thảo mã hiển thị khi bạn soạn thảo mã nguồn.
- **Thanh mini Build:** lệnh xây dựng và chạy.

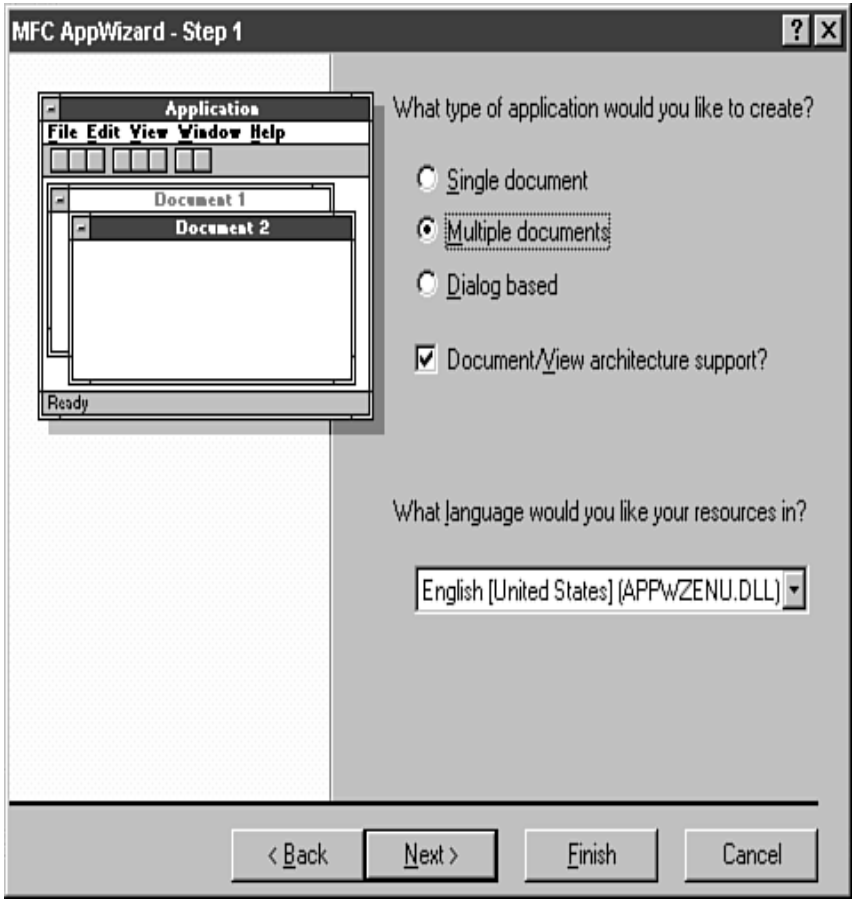
Tạo chương trình bằng AppWizard

- **Microsoft Visual C++ chọn menu File - New**
  - **MFC AppWizard (exe)**
  - **Project Name: điền tên dự án vào**
  - **Location: nơi chứa dự án.**



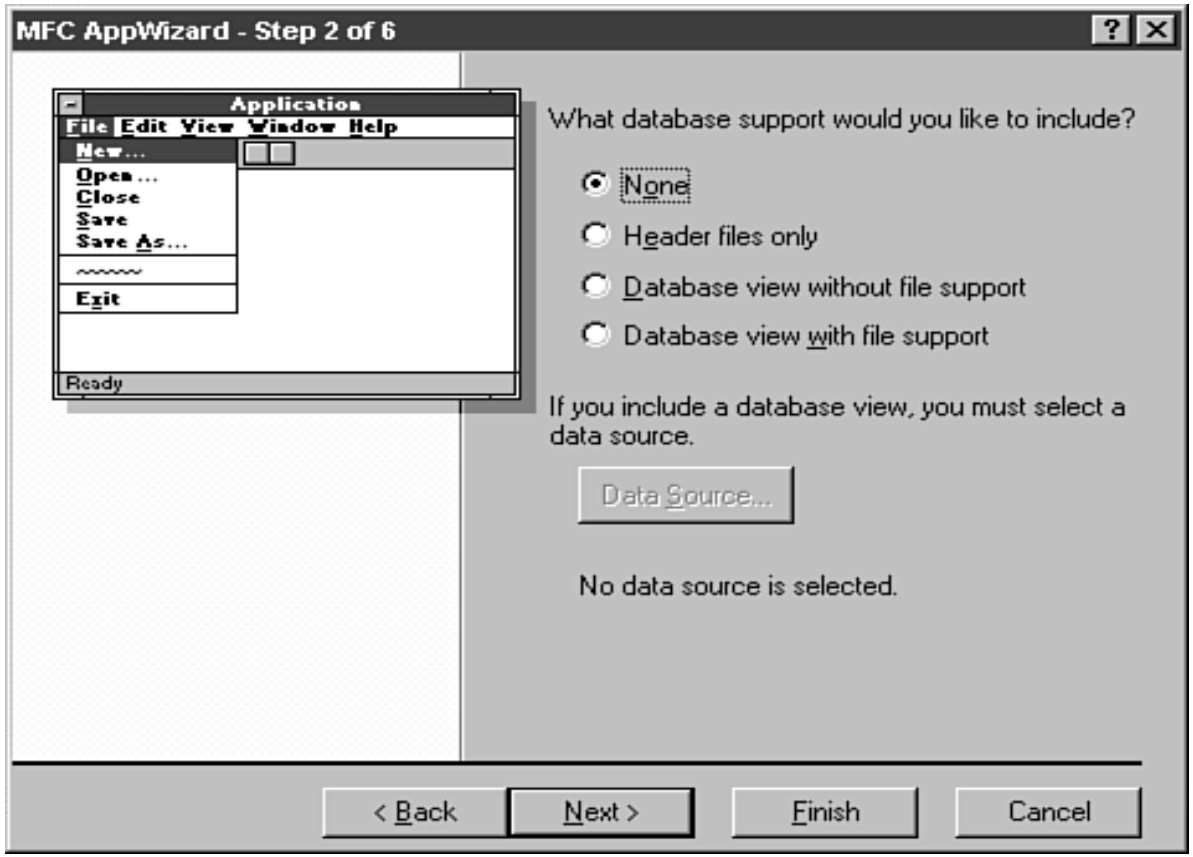
Tạo chương trình bằng AppWizard

- **Bước 1: Có 3 kiểu ứng dụng được chọn tại bước này:**
  - **Single document: một tài liệu trong một lúc.**
  - **Multiple document: nhiều tài liệu cùng một lúc để làm việc.**
  - **Dialog based: hộp hội thoại làm giao diện người sử dụng.**



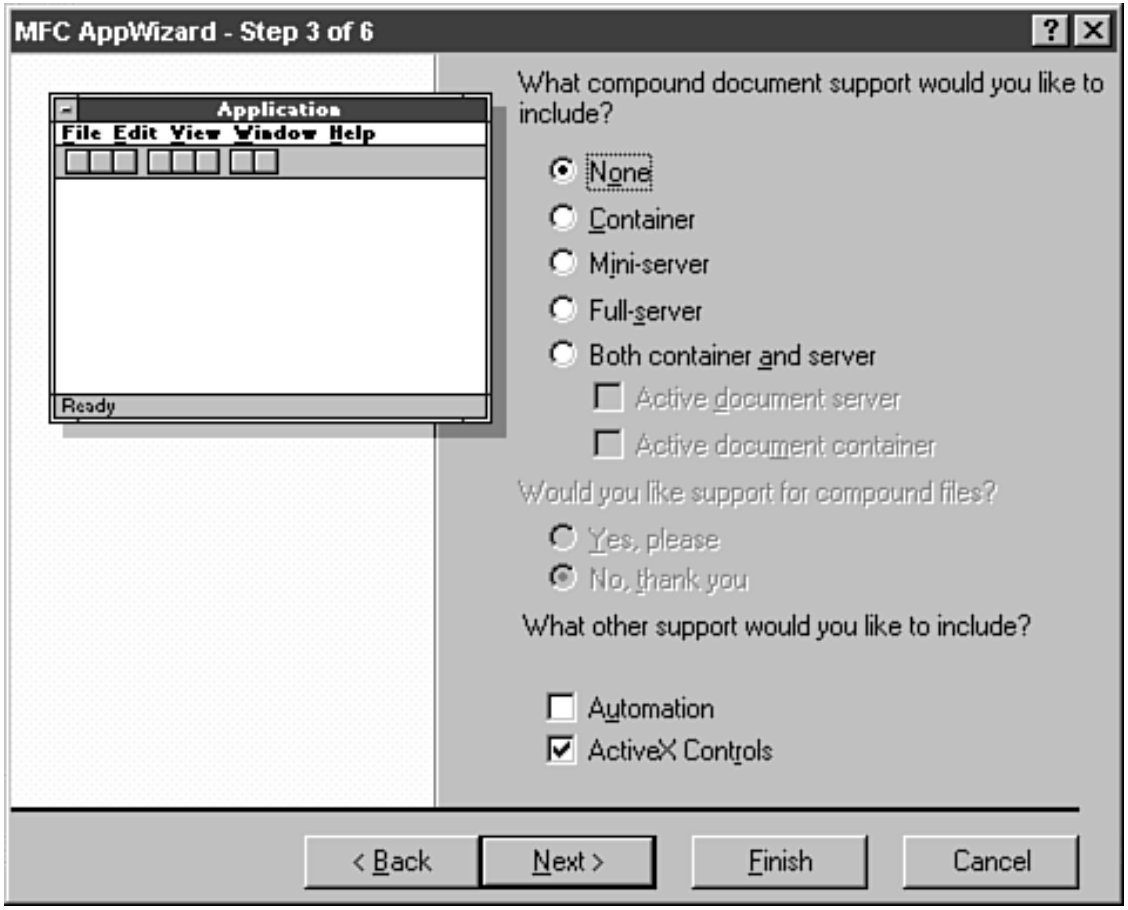
Tạo chương trình bằng AppWizard

❖ **Bước 2**: cho phép chọn mức hỗ trợ cơ sơ dữ liệu:



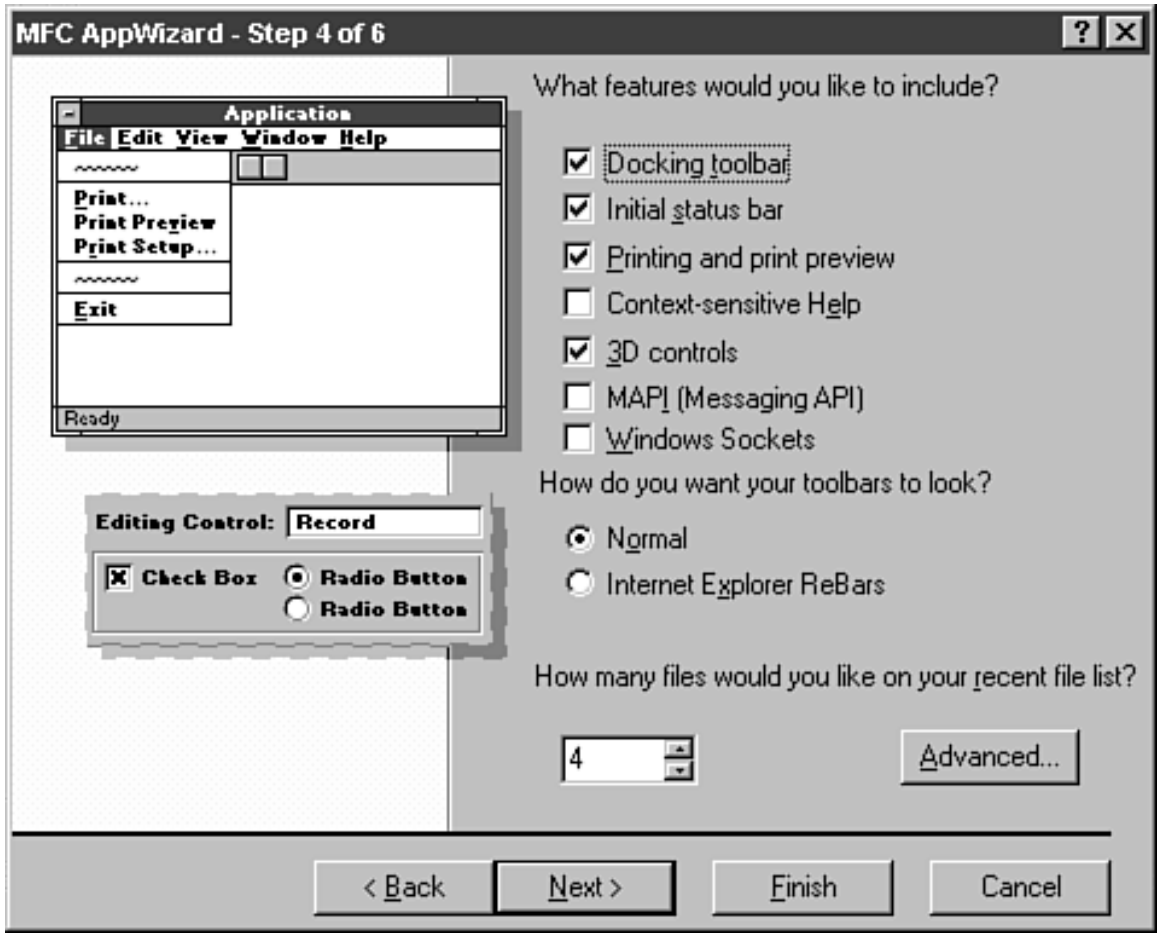
Tạo chương trình bằng AppWizard

- **Bước 3**: chọn tài liệu đa hợp hỗ trợ mà chương trình cần:



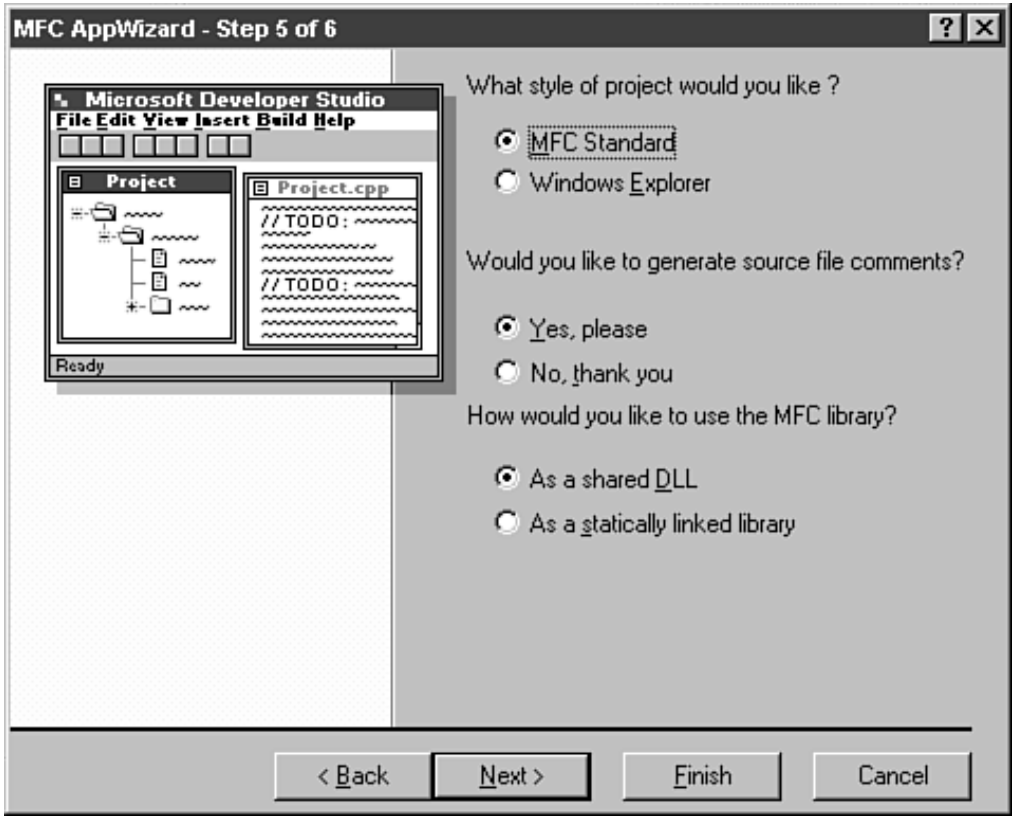
Tạo chương trình bằng AppWizard

❖ **Bước 4: Chọn một số đặc tính cho giao diện.**



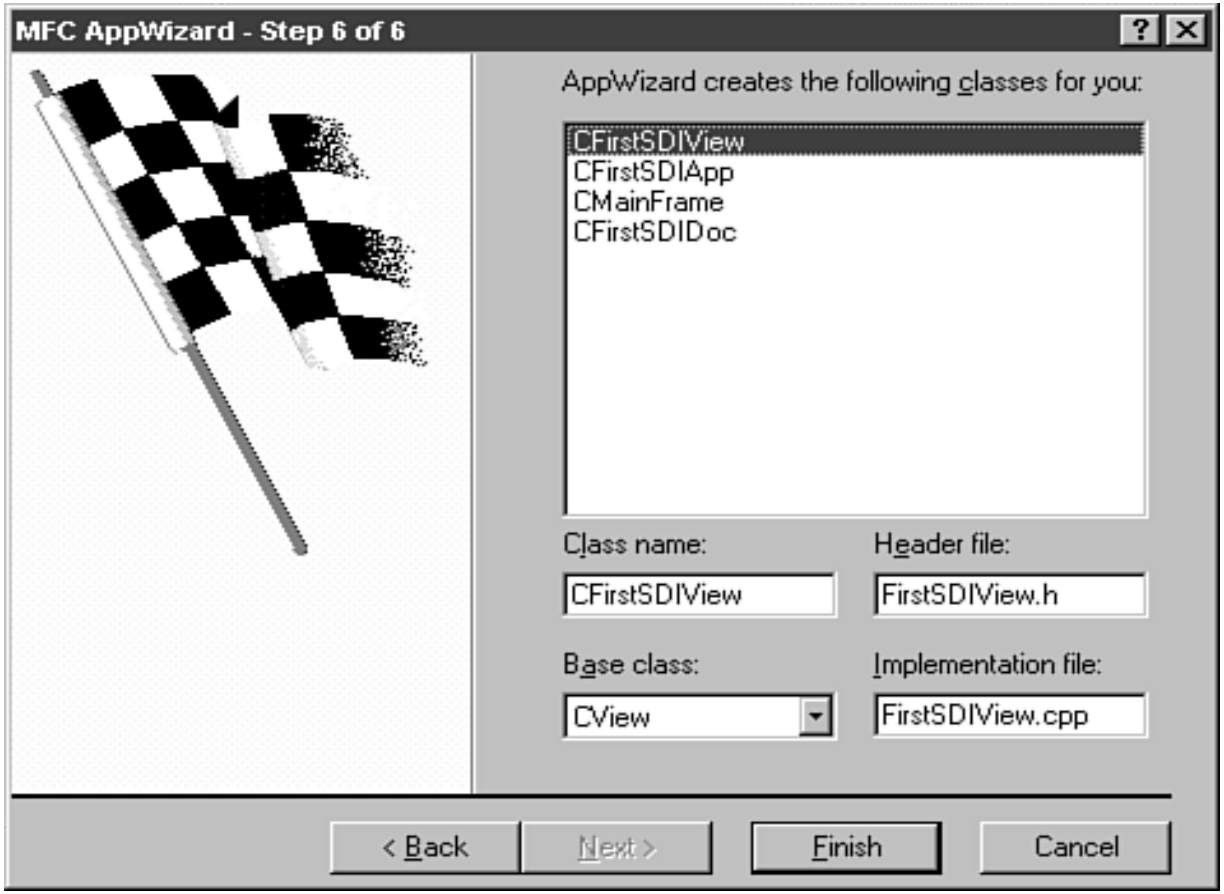
Tạo chương trình bằng AppWizard

- **Bước 5: Chọn những ghi chú và những thư viện MFC trong chương trình.**



Tạo chương trình bằng AppWizard

❖ **Bước 6: Chứng thực filenames và classname.**



Phân tích các tập tin của một ứng dụng

- **Ứng dụng Single Document**
  - Đối tượng dẫn xuất từ **4** lớp đối tượng cơ sở **Application classes**, **Document classes**, **ViewClasses** và **Frames classes** .
  - Ví dụ:

Lớp cơ sở	**Lớp dẫn xuất**	**Tên tập tin**
CWinApp	Cvd1App	Vd1.cpp
CDocument	Cvd1Doc	Vidu1Doc.cpp
CView	Cvd1View	Vidu1View.cpp
CFrameWnd	CmainFrame	MainFrm.cpp
CMDIChildWnd	CchildFrame	ChildFrm.cpp

Phân tích các tập tin của một ứng dụng

- **Tập tin RESOURCE.H:** chứa toàn bộ những lệnh **#define** dùng định nghĩa chỉ danh ID những tài nguyên như là:
  - **DD_ABOUTBOX**: Đối với tài nguyên là một hộp hội thoại about box
  - **IDR_MAINFRAME**: Đối với việc chia sẻ sử dụng bởi nhiều loại tài nguyên.
  - **IDR_VIDU1TYPE**: Đối với tài nguyên là biểu tượng tài liệu

Phân tích các tập tin của một ứng dụng

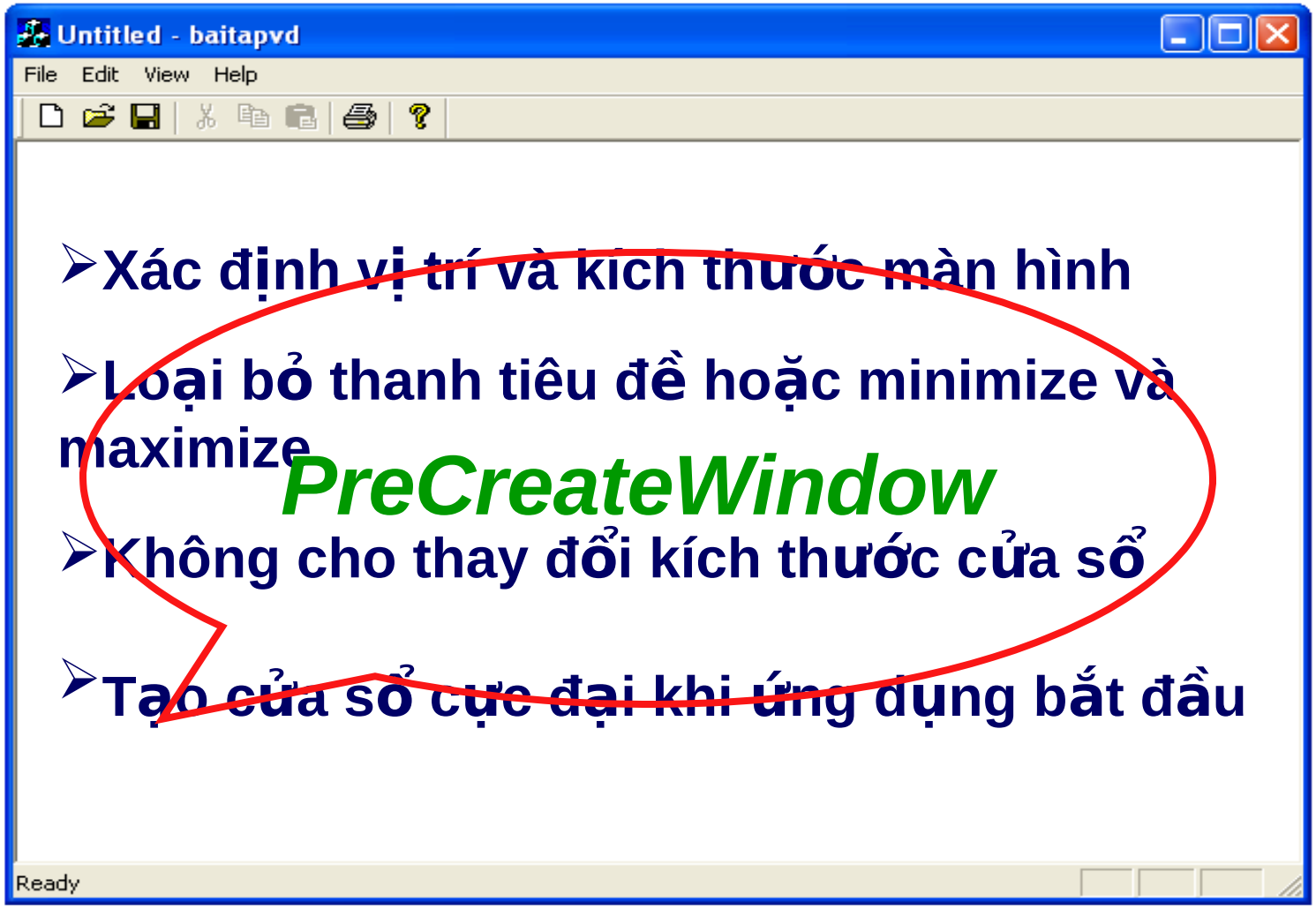
- **Tập tin Vidu1.H:** chứa các khai báo liên quan đến lớp Cvidu1App:
- **Tập tin MainFrm.H**: chứa các khai báo liên quan đến lớp khung cửa sổ chính (main frame) của ứng dụng.
- **Tập tin Vidu1Doc.H:** chứa các khai báo liên quan đến tài liệu.
- **Tập tin Vidu1.RC:** chứa các tài nguyên liên quan đến menu, phím nóng, hộp hội thoại,…
- **Tập tin Vidu1.CPP:** định nghĩa các hàm đã khai báo trong lớp CVidu1App, là lớp chương trình chính.

Thực đơn, thanh công cụ và thanh trạng thái

- Màn hình ứng dụng
- Thực đơn (Menu)
- Thanh công cụ (ToolBar)
- Thanh trạng thái (Status bar)

Thực đơn, thanh công cụ và thanh trạng thái

❖ **Màn hình ứng dụng:**



Xác định vị trí và kích thước màn hình

- Để đặt ứng dụng giữa màn hình và chiếm 90% màn hình:

```
int xSize= GetSystemMetrics(SM_CXSCREEN);
int ySize= GetSystemMetrics(SM_CYSCREEN);
cs.cx = xSize *9/10;
cs.cy = ySize *9/10;
cs.x   = (xSize – cs.cx )/2;
cs.y   = (ySize – cs.cy )/2;
```

- Trong đó: cs là biến cấu trúc *CREATESTRUCT*

Màn hình ứng dụng

- **Loại bỏ minimize và maximize:**

  cs.style &= ~(WS_MAXIMIZEBOX | WS_MINIMIZEBOX);

- **Không cho thay đổi kích thước cửa sổ:**

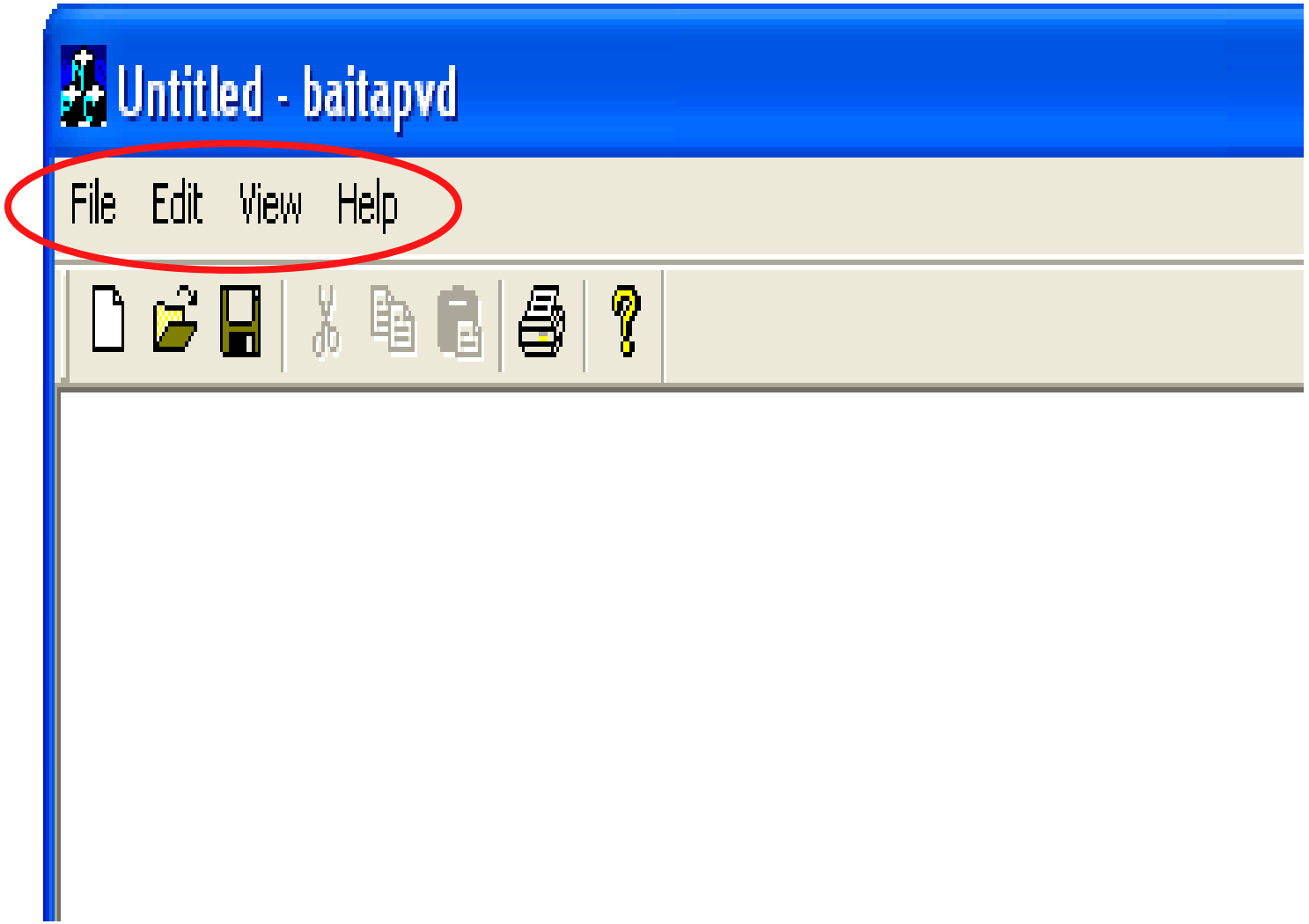
  cs.style &= ~WS_THICKFRAME;

- **Tạo cửa sổ cực đại khi ứng dụng bắt đầu:**

  Trong hàm **ShowWindow()** thay đổi cờ m_nCmdShow thành cờ SW_SHOWMAXIMIZED:

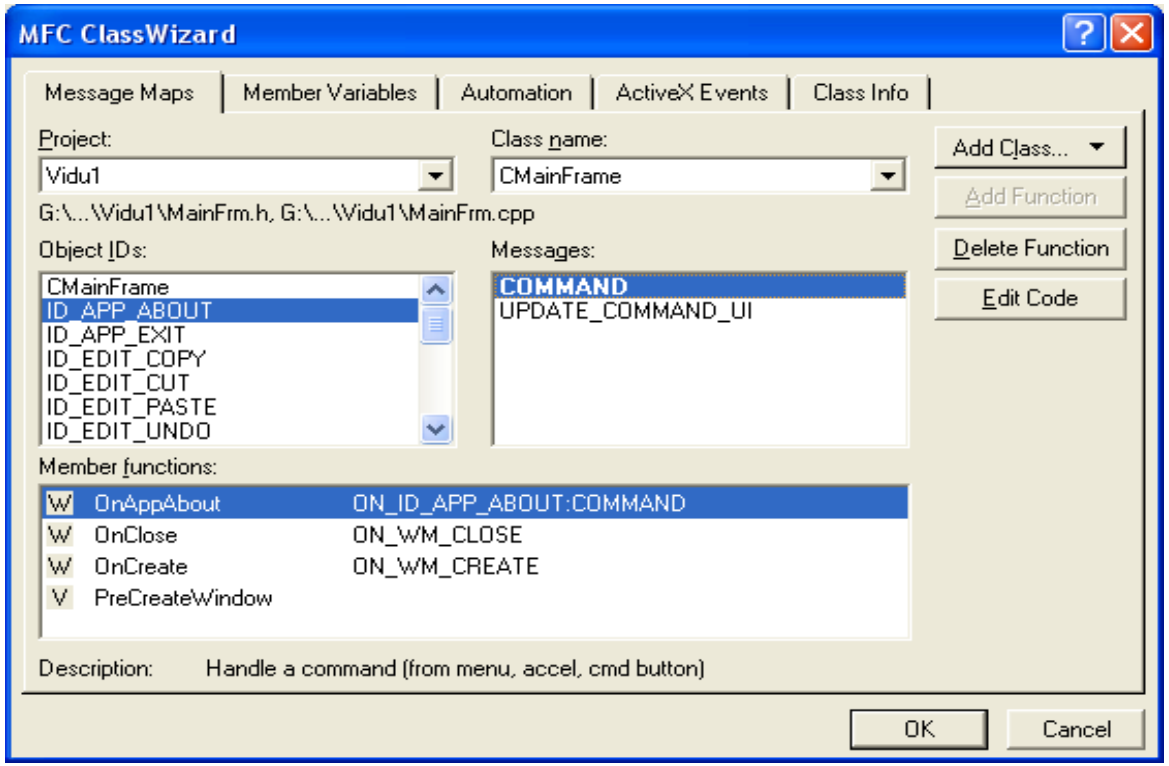
  m_pMainWnd-> ShowWindow(SW_SHOWMAXIMIZED);

Thực đơn (Menu)



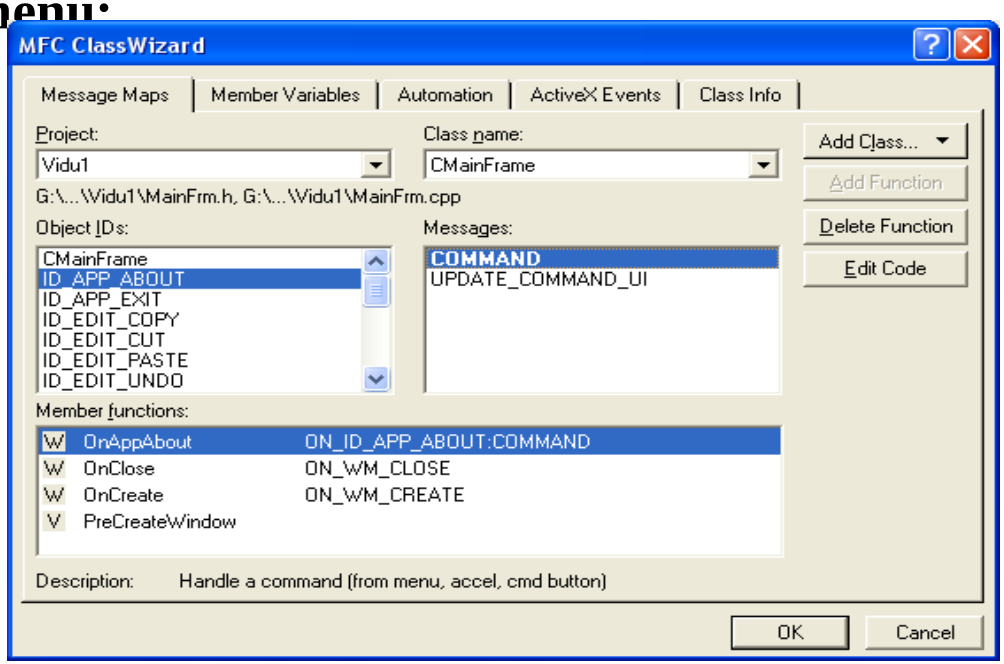
Thực đơn (Menu)

- **Sử dụng ClassWizard để tự động thêm một bộ phận điều khiển lệnh menu:** là một hàm thành viên của lớp. Hàm thành viên này được thi hành khi người sử dụng kích hoạt vào menu.



Các bước tạo

- Chọn View/ClassWizard – Hộp hội thoại hiện ra và chọn lớp muốn xử lý lệnh menu:



- Chọn ID mục menu cần xử lý trong khung Object Ids và chọn COMMAND trong khung Message.
- ClassWizard tự tạo hàm xử lý trống trong khung Member function.
- Click vào nút Edit Code để chuyển đến hàm xử lý. Từ đây thêm vào đoạn mã xử lý cho lệnh menu.

Thực đơn (Menu)

- Ví dụ: Tạo một mục menu, với yêu cầu click vào mục menu đó thi sẽ xuất ra một thông báo.

```
void CMainFrame::Tên_Hàm_Xử_Lý_Menu()
{
  // TODO: Add your command handler code here
    AfxMessageBox(“Thông báo", MB_YESNO);
}
```

Thực đơn (Menu)

❖ **Các lệnh menu có hiệu lực và không có hiệu lực:**

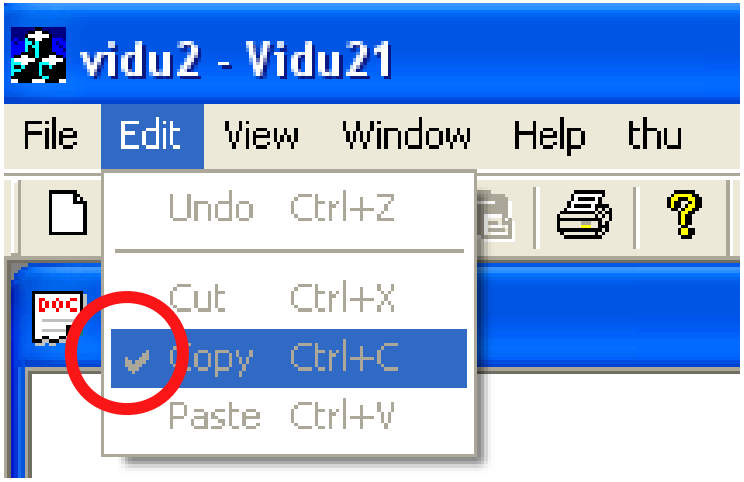
- **Khi sử dụng ClassWizard để thêm mới một mục menu thay vì chọn COMMAND trong khung Messages thì chọn UPDATE_COMMAND_UI và trong hàm xử thông điệp bổ sung câu lệnh:**

```
pCmdUI->Enable(m_bWzd);
```

- **Nếu m_bWzd =TRUE thì menu trở nên hoạt động,**
- **Nếu m_bWzd = FALSE có kết quả ngược lại.**

Thực đơn (Menu)

❖ **Đánh dấu kiểm tra các lệnh menu:**



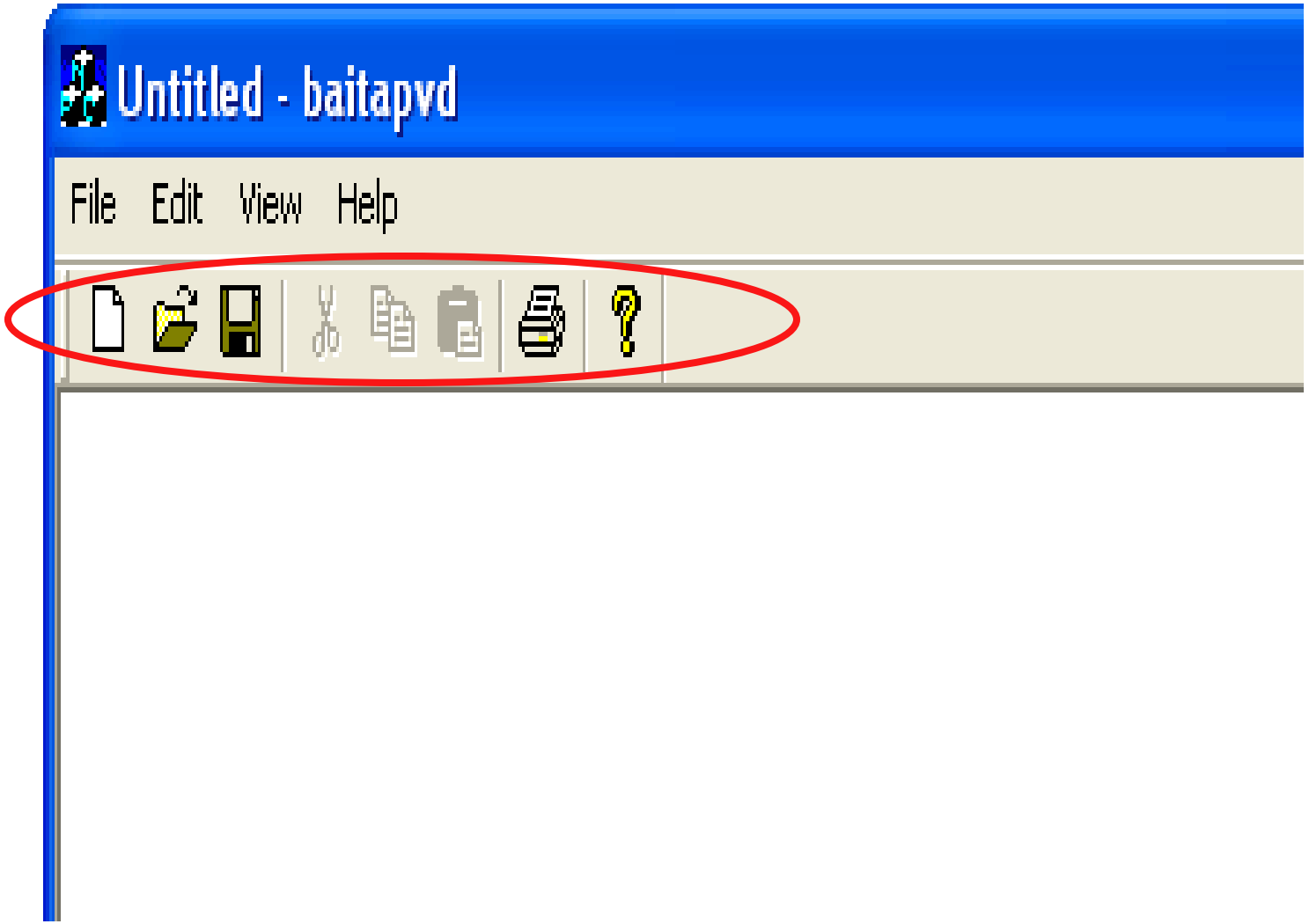
- **Thực hiện bằng ClassWizard thêm hàm xử lý chọn UPDATE_COMMAND_UI và viết lệnh xử lý:**

  pCmdUI->SetCheck(m_bWzd);

  m_bWzd = TRUE có dấu kiểm tra

  m_bWzd = FALSE bỏ dấu kiểm tra.

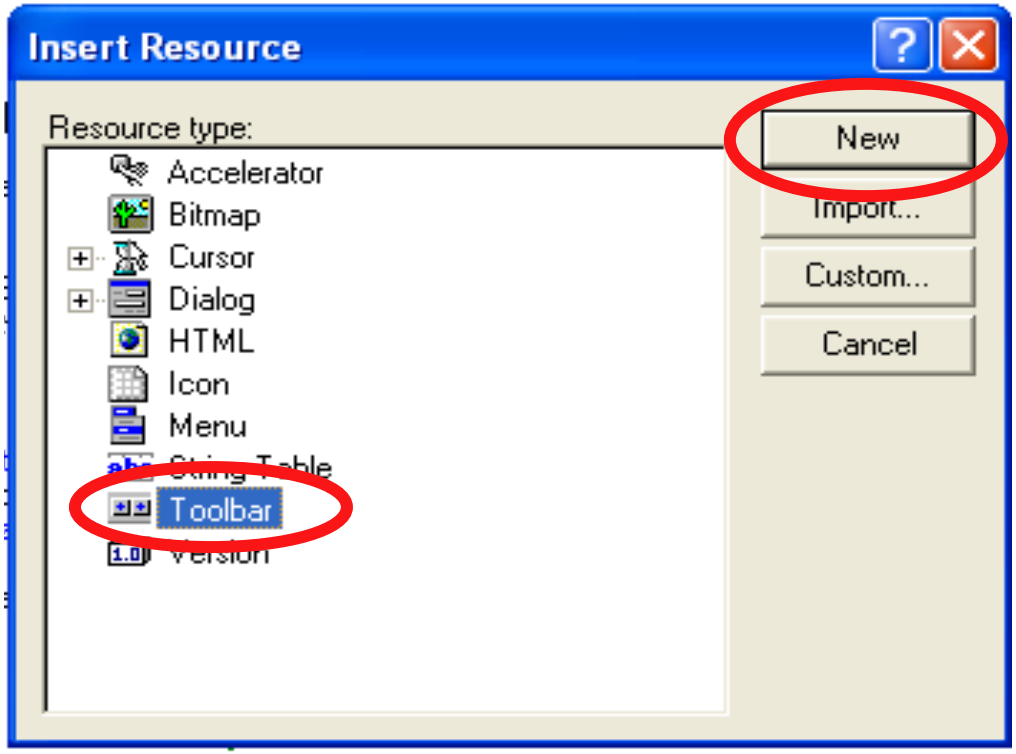
Thanh công cụ (ToolBar)



Thanh công cụ (ToolBar)

❖ **Thêm thanh công cụ mới vào ứng dụng:**

- **Bước 1: Soạn thảo (tạo) thanh công cụ**
  - **Click vào menu Insert/Resource, chọn toolbar từ danh sách và click vào new**



Thanh công cụ (ToolBar)

- **Thêm thanh công cụ mới vào ứng dụng:**
  - **Bước 2: khai báo đối tượng ToolBar thuộc lớp CToolBar trong thành phần protected của lớp CframeWnd.**

Cú pháp:

CToolBar Tên_đối_tượng_ToolBar;

Thanh công cụ (ToolBar)

❖ **Thêm thanh công cụ mới vào ứng dụng:**

- **Bước 3: Viết mã lệnh trong hàm Create() của lớp CMainFrame .**

```
if (!toolbar.Create(this) || !toolbar.LoadToolBar(ID_TOOLBAR))
  {
        TRACE0("Failed to create toolbar\n");
        return -1;      // fail to create
  }
  toolbar.SetBarStyle(oolbar.GetBarStyle() | CBRS_TOOLTIPS |
  CBRS_FLYBY | CBRS_SIZE_DYNAMIC);
  oolbar.EnableDocking(CBRS_ALIGN_ANY);
```

Thanh công cụ (ToolBar)

- **Các thanh công cụ thêm vào sẽ được sắp xếp theo hàng:**

*DockControlBar(CControlBar *pBar, UINT nDockBarID= 0)*;

pBar: con trỏ đối tượng toolbar

nDockBarID: gtrị xác định vị trí thanh công cụ

- **AFX_IDW_DOCKBAR_TOP : Toolbar xếp vào phía trên vùng client cửa sổ**
- **AFX_IDW_DOCKBAR_BOTTOM : Toolbar xếp vào phía dưới vùng client cửa sổ**
- **AFX_IDW_DOCKBAR_LEFT : Toolbar xếp vào phía bên trái vùng client cửa sổ**
- **AFX_IDW_DOCKBAR_RIGHT : Toolbar xếp vào phía bên phải vùng client cửa sổ**

Thanh công cụ (ToolBar)

- **Làm có hiệu lực, mất hiệu lực các nút thanh công cụ :**
  - **Sử dụng ClassWizard , chọn ID của nút công cụ – UPDATE_COMMAND_UI**

```
void CVd2View::OnUpdateButton(CCmdUI* pCmdUI)
{
    pCmdUI->Enable(m_bWzd);
    // TODO: Add your command update UI handler code here
}
```

- Nếu m_bWzd = TRUE thì có hiệu lực

- Ngược lại m_bWzd = FALSE thì không có hiệu lực.

Thanh công cụ (ToolBar)

- **Thêm text vào nút :**
  - ***phương thức :***

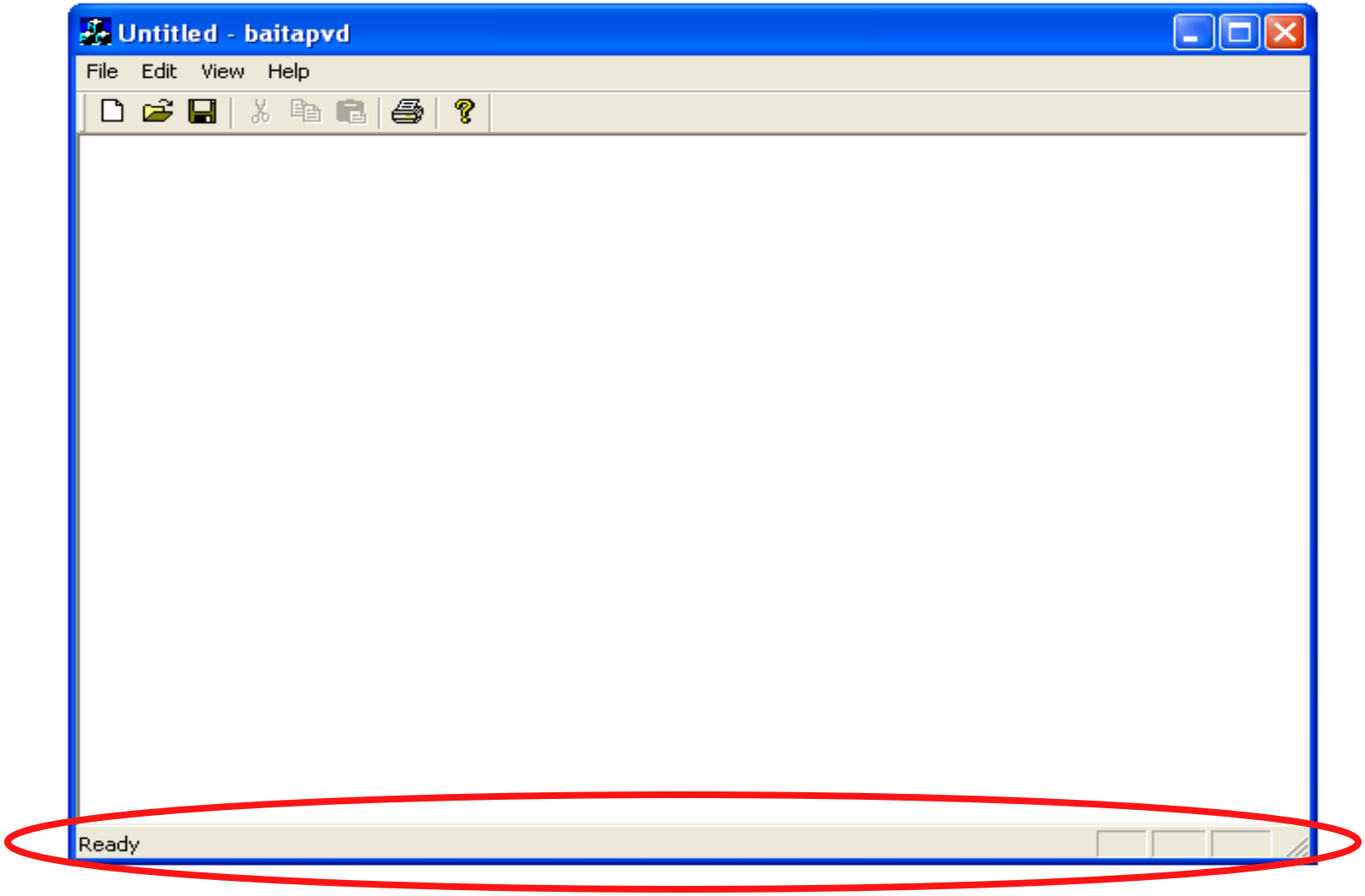
    SetButtonText(int nIndex, LPCTSTR lpszText);
    - ***nIndex: số thứ tự của nút thêm text***
    - ***lpszText: chuỗi văn bản thêm vào***
  - ***Xác định kích thước các nút trong toolbar và kíchước iMage***

    SetSizes(SIZE sizeButton, SIZE sizeImage);

Ví dụ:

```
ToolBar.SetButtonText (9, _T ("Print"));
ToolBar.SetSizes (CSize (48, 42), CSize (40, 19));
```

Thanh trạng thái (Status bar)



Thanh trạng thái (Status bar)

- **Thêm thanh trạng thái mới vào ứng dụng:**
  - **khai báo đối tượng m_wndStatusBar thuộc lớp CStatusBar trong thành phần protected của lớp CframeWnd.**
  - **Viết mã lệnh trong hàm Create**

```
if (!m_wndStatusBar.Create(this) ||!
   m_wndStatusBar.SetIndicators(indicators,
   sizeof(indicators)/sizeof(UINT)))
  {
            TRACE0("Failed to create status bar\n");
            return -1;      // fail to create
  }
```

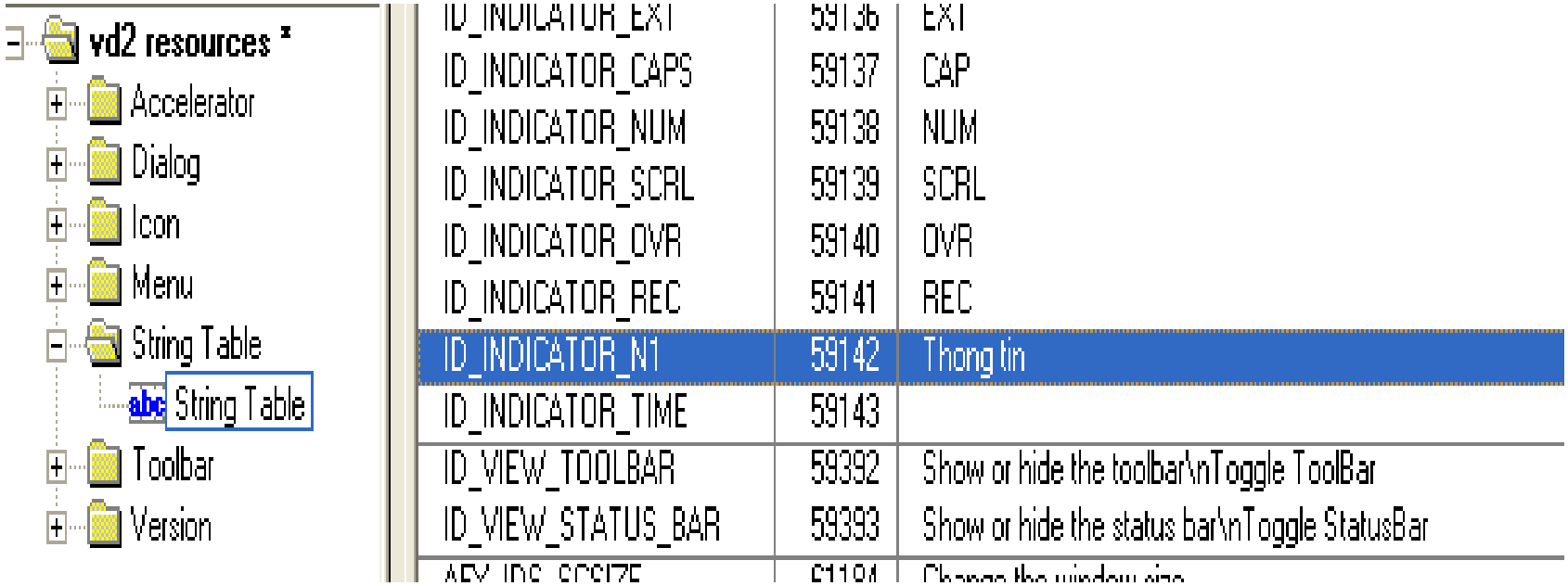
Thanh trạng thái (Status bar)

- **Hàm SetIndicator đặt các chỉ danh indicator vào thanh trạng thái, số indicator và indicator phải khai báo như sau trong đầu tập tin MainFrm.cpp**

```
static UINT indicators[] =
{
    ID_SEPARATOR,          // status line indicator
    ID_INDICATOR_CAPS,
    ID_INDICATOR_NUM,
    ID_INDICATOR_N1,
    ID_INDICATOR_TIME,
};
```

Thanh trạng thái (Status bar)

Mỗi ID_INDICATOR tương ứng với một chuỗi trạng thái cần hiển thị và được khai báo trong tập tin tài nguyên String table



Thanh trạng thái (Status bar)

- **Để hiện thị các thông tin lên indicator trên thanh trạng thái cần bổ sung hàm:**

void CMainFrame:: OnUpdateIndicatorN1(CCmdUI* pCmdUI)

{

// TODO: Add your command update UI handler code here

pCmdUI->Enable(TRUE);

}

Và bổ sung khai báo trong tập tin MainFrm.h

```
// Generated message map functions
protected:
    //{{AFX_MSG(CVd2View)
    afx_msg void OnViewToolbar();
    afx_msg void OnUpdateButton32774(CCmdUI* pCmdUI);
    afx_msg void OnUpdateIndicatorN1(CCmdUI* pCmdUI);
    afx_msg void OnButton32774();

    //}}AFX_MSG
    DECLARE_MESSAGE_MAP()
};
```

Thanh trạng thái (Status bar)

- ***Khai báo bảng thông điệp trong tập tin MainFrm.cpp***

```
BEGIN_MESSAGE_MAP(CMainFrame, CFrameWnd)
    //{{AFX_MSG_MAP(CMainFrame)
    ON_WM_CREATE()
    ON_WM_CLOSE()
    ON_COMMAND(ID_GT_CN, OnGtCn)
    ON_UPDATE_COMMAND_UI(ID_GT_CN, OnUpdateGtCn)
    ON_COMMAND(ID_GT_CQ, OnGtCq)
    ON_UPDATE_COMMAND_UI(ID_GT_CQ, OnUpdateGtCq)
    ON_COMMAND(ID_NEW, OnNew)
    ON_COMMAND(ID_BUTTON32774, OnButton32774)
    ON_COMMAND(ID_BUTTON32775, OnButton32775)
    ON_COMMAND(ID_VIEW_TOOLBAR, OnViewToolbar)
    ON_UPDATE_COMMAND_UI(ID_VIEW_TOOLBAR, OnUpdateViewToolbar)
    ON_WM_TIMER()
    ON_UPDATE_COMMAND_UI (ID_INDICATOR_TIME, OnUpdateTime)
    ON_UPDATE_COMMAND_UI(ID_INDICATOR_N1, OnUpdateIndicatorN1)
    //}}AFX_MSG_MAP


END_MESSAGE_MAP()
```

ĐỒ HỌA VÀ XỬ LÝ CÁC THÔNG ĐIỆP ĐẦU VÀO

- Các hàm đồ họa cơ sở
- Xử lý văn bản
- Xửlý sự kiện chuột (mouse)
- Xử lý sự kiện phím (keyboard)

ĐỒ HỌA VÀ XỬ LÝ CÁC THÔNG ĐIỆP ĐẦU VÀO

❖ Device context:

- **Device context** là một cấu trúc dữ liệu được duy trì để kết hợp với một thiết bị hiển thị như màn hình, máy in,…
- Khi vẽ đầu tiên phải khai báo **device context** và sau khi sử dụng thiết bị **device context** phải giải phóng **device context**.

  - ***CDC * pDC = GetDC();***; *// Khai báo thiết bị DC.*
  - ***ReleaseDC(hWnd, hDC)***; *// Giải phóng DC*

ĐỒ HỌA VÀ XỬ LÝ CÁC THÔNG ĐIỆP ĐẦU VÀO

- Hàm **LineTo**(): **Vẽ đường thẳng**

  pDC -> LineTo(int X, int Y);

  Vẽ đường thẳng từ vị trí hiện hành đến tọa độ X, Y

- Hàm **MoveToEx**(): **Thiết đặt vị trí hiện hành**

  pDC -> MoveToEx(int X, int Y);

  Tọa độ hiện hành được xác định bởi X, Y

ĐỒ HỌA VÀ XỬ LÝ CÁC THÔNG ĐIỆP ĐẦU VÀO

- ***Vẽ hình chữ nhật:***

  pDC -> Rectangle*(int **upX**, int **upY**, int **lowX**, int **lowY**)*

  - **upX, upY:** **góc trên bên trái**
  - **lowX, lowY:** **góc dưới bên phải**

❖ *Vẽ Elip:*

pDC->Ellipse(int upX, int upY, int lowX, int lowY)

- **upX, upY: góc trên bên trái**
- **lowX, lowY: góc dưới bên phải**

❖ *Vẽ Pie:*

*pDC->pie(int **upX**, int **upY**, int **lowX**, int **lowY,** int **Xs,** int **Ys,** int **Xe,** int **Ye**)*

- **upX, upY: góc trên bên trái**
- **lowX, lowY: góc dưới bên phải**
- ***Xs, Ys: Điểm bắt đầu cung***
- ***Xe, Ye: Điểm kết thúc cung***

ĐỒ HỌA VÀ XỬ LÝ CÁC THÔNG ĐIỆP ĐẦU VÀO

- ***Vẽ đường liền kề nhau:***

  pDC->Polyline(POINT pt[], int n);

  - **pt: mảng chứa các điểm tọa độ**
  - **n : số lượng tọa độ**

ĐỒ HỌA VÀ XỬ LÝ CÁC THÔNG ĐIỆP ĐẦU VÀO

- ***Tạo thuộc tính bút vẽ:***

CPen pen(int style, int width, COLOREF color);

- **Style: kiểu bút vẽ và có các giá trị:**
  - PS_DASH : Gạch ngang
  - PS_DASHDOT : Gạch ngang chấm
  - PS_DASHDOTDOT : Gach ngang chấm chấm
  - PS_DOT : Chấm
  - PS_SOLID : Đường thẳng liền nét.
- **Width: độ dày của bút vẽ**
- ***Color: RGB(R, G, B)***

CPen *pPen = pDC->SelectObject(&pen) //thiết lập

ĐỒ HỌA VÀ XỬ LÝ CÁC THÔNG ĐIỆP ĐẦU VÀO

- ***Tạo chổi quét:***
  - ***Cbrush Brush (COLOREF color)***
    - ***Color: RGB(R, G, B)***

- ***Thiết lập chổi quét***
  - ***CBrush *pBrush = pDC->SelectObject (&Brush)***

❖ Hàm TextOut:

pDC->TextOut(x, y, string, strLength);

Trong đó:

- **pDC** là device context của cửa sổ cần xuất.
- **x, y** điểm bắt đầu của chuỗi ký tự trong client
- **string** là một con trỏ trỏ đốn một chuỗi ký tự.
- **strLength** là số ký tự trong chỗi cần xuất.

Ví dụ:

- CString string(“This is text”);
- pDC->TextOut(x, y, str, str.GetLength());
- pDC-> SetTextAlign(TA_CENTER) // TA_RIGHT

ĐỒ HỌA VÀ XỬ LÝ CÁC THÔNG ĐIỆP ĐẦU VÀO

- **Thiết lập màu chữ**
  - **pDC-> SetTextColor(*COLOREF color*)**
- **Thiết lập màu nền**
  - **pDC-> SetBkColor(*COLOREF color*)**

ĐỒ HỌA VÀ XỬ LÝ CÁC THÔNG ĐIỆP ĐẦU VÀO

- **Font: có các thuộc tính**
  - **Typeface: loại font (Times, Courier, Arial, …)**
  - **Style: kiểu dáng (normal, thin, bold,…)**
  - **Size: kích cỡ chữ, được xác định theo đơn vị point, 1 point = 1/72 inch = 0.013837 inch**

ĐỒ HỌA VÀ XỬ LÝ CÁC THÔNG ĐIỆP ĐẦU VÀO

- **Thiết lập font chữ cho vùng client**
  - ***1. CFont Font; //khai báo đối tượng Font***
  - **2**. Gọi **Font.CreateFont** để tạo ra font có thuộc tính xác định

```
Font.CreateFont(int Height, //kích thước điểm
                0, 0, 0,
                FW_NORMAL, // Độ dầy, có thể là FW_BOLD
                0,          // nếu 1  nghiêng
                0,          // Nếu 1 gạch dưới
                0,          //nếu 1 gạch ngang
                0,0, 0, 0,0,"VNI-Times"); //kiểu chữ
```

  - **3**. Gọi **SelectObject**() cài đặt Font vào vùng client.

```
CFont *pFont = (CFont *) pDC->SelectObject(&font);
SetFont(pFont); //thiết lập
```

ĐỒ HỌA VÀ XỬ LÝ CÁC THÔNG ĐIỆP ĐẦU VÀO

- **Thuật vẽ bitmap**
  - **Tạo nguồn bitmap trong tập tin tài nguyên** *:*
    - ***ID_BITMAP BITMAP "c:\filebitmap.bmp"***
  - **Nạp bitmap từ nguồn tài nguyên**
    - CBitmap bitmap; *//khao báo*
    - bitmap.LoadBitmap(IDB_BITMAP);
  - ***Đạt được DC của bộ nhớ sẽ chứa bitmap***
    - CDC DcComp;
    - DcComp.CreateCompatibleDC(pDC);
  - ***Chọn bitmap trong device context của bộ nhớ.***
    - DcComp.SelectObject(bitmap);

Thuật vẽ bitmap (TT)

- **Nhận thông tin của BitMap**
  - BITMAP bmpinfo; *//khao báo*
  - bitmap.GetObject(sizeof(bmpinfo), &bmpinfo));
- ***Sao chép bitmap từ DC của bộ nhớ sang DC của cửa sổ***
  - *PDC->BitBlt(x,y, bmpinfo.bmWidth, bmpinfo.bmHeight, &DcComp,0,0,SRCCOPY);*

CÁC THÔNG ĐIỆP SỰ KIỆN CHUỘT

- **Các thông điệp MOUSE trong vùng client**
  - ***Khi mouse di chuyển trên màn hình luôn phát ra thông điệp WM_MOUSEMOVE và khi các nút mouse được nhấn hay thả trong vùng client của một cửa sổ hàm window nhận các thông điệp:***

Nút	Nhấn	Nhả	Nhấn (lần 2)
Trái	WM_LBUTTONDOWN	WM_LBUTTONUP	WM_LBUTTONDBLCLK
Giữa	WM_MBUTTONDOWN	WM_MBUTTONUP	WM_MBUTTONDBLCLK
Phải	WM_RBUTTONDOWN	WM_RBUTTONUP	WM_RBUTTONDBLCLK

CÁC THÔNG ĐIỆP SỰ KIỆN CHUỘT

- **Hàm xử lý thông điệp MOUSE:**
  - **Lớp :: Thông_Điệp (UINT nFlags, CPoint point)**
  - **Trong đó:**
    - **nFlags: Cờ xác định nút chuột được nhấn hay nhả và nút đó là nút nào.**
      - MK_LBUTTON: Nút trái được nhấn
      - MK_MBUTTON: Nút giữa được nhấn
      - MK_RBUTTON: Nút phải được nhấn
    - **Point: Tọa độ xác định vị trí chuột**

CÁC THÔNG ĐIỆP SỰ KIỆN PHÍM

- Thông điệp bàn phím mà chương trình ứng dụng nhận được do **Windows** chuyển đến phân biệt tình huống phím gõ hay ký tự gõ.

- Các phím Shift, phím chức năng(F1, F2, ….), các phím mũi tên di chuyển, các phím insert, Delete,… phát sinh phím gõ mà không phát sinh thông điệp ký tự gõ.

CÁC THÔNG ĐIỆP SỰ KIỆN PHÍM

- **Thông điệp phím gõ**
  - Khi một phím được nhấn Windows đặt thông điệp **WM_KEYDOWN** và đưa vào hàng đợi thông điệp của ứng dụng hay cửa sổ nhận được sự quan tâm
  - Khi nhả phím Windows đặt thông điệp **WM_KEYUP** sẽ được hệ điều hành phân phát tới hàng đợi đó

CÁC THÔNG ĐIỆP SỰ KIỆN PHÍM

- **Mã phím ảo:** Cho biết giá trị của phím nhấn hay thả.

Giá trị thập phân	Giá trị Hex	Mã phím ảo	Phím
08	08	VK_BACK	Backspace
09	09	VK_TAB	Tab
13	0D	VK_RETURN	Enter
16	10	VK_SHIFT	Shift trái , phải
17	11	VK_CONTROL	Ctrl trái, phải
18	12	VK_MENU	Alt trái, phải
19	13	VK_PAUSE	Pause
20	14	VK_CAPITAL	Caps Lock
27	1B	VK_ESCAPE	Esc
32	20	VK_SPACE	Space bar
33	21	VK_PRIOR	Page up
34	22	VK_NEXT	Page down

CÁC THÔNG ĐIỆP SỰ KIỆN PHÍM

35	23	VK_END	End
36	24	VK_HOME	Home
37	25	VK_LEFT	Mũi tên trái
38	26	VK_UP	Mũi tên lên
39	27	VK_RIGHT	Mũi trón phải
40	28	VK_DOWN	Mũi tên xuống
45	2D	VK_INSERT	Insert
46	2E	VK_DELETE	Delete
112 –123	70- 7B	VK_F1,.... VK_F12	Phím chức năng
144	90	VK_NUMLOCK	Numlock
145	91	VK_SCROLL	Scroll Lock

CÁC THÔNG ĐIỆP SỰ KIỆN CHUỘT

- **Hàm xử lý thông điệp PHÍM:**
  - **Lớp :: Thông_Điệp (UINT nChar, UINT nRepCnt UINT nFlags)**
  - **Trong đó:**
    - **nChar: Phím được nhấn**
    - **nRepCnt: Số lần nhấn phím.**
      - **Nếu nhả thì giá trị = 1.**
    - **nFlags: Cờ xác định có kèm theo phím mở rộng được nhấn hay không.**

BỘ ĐỊNH THỜI TIMER

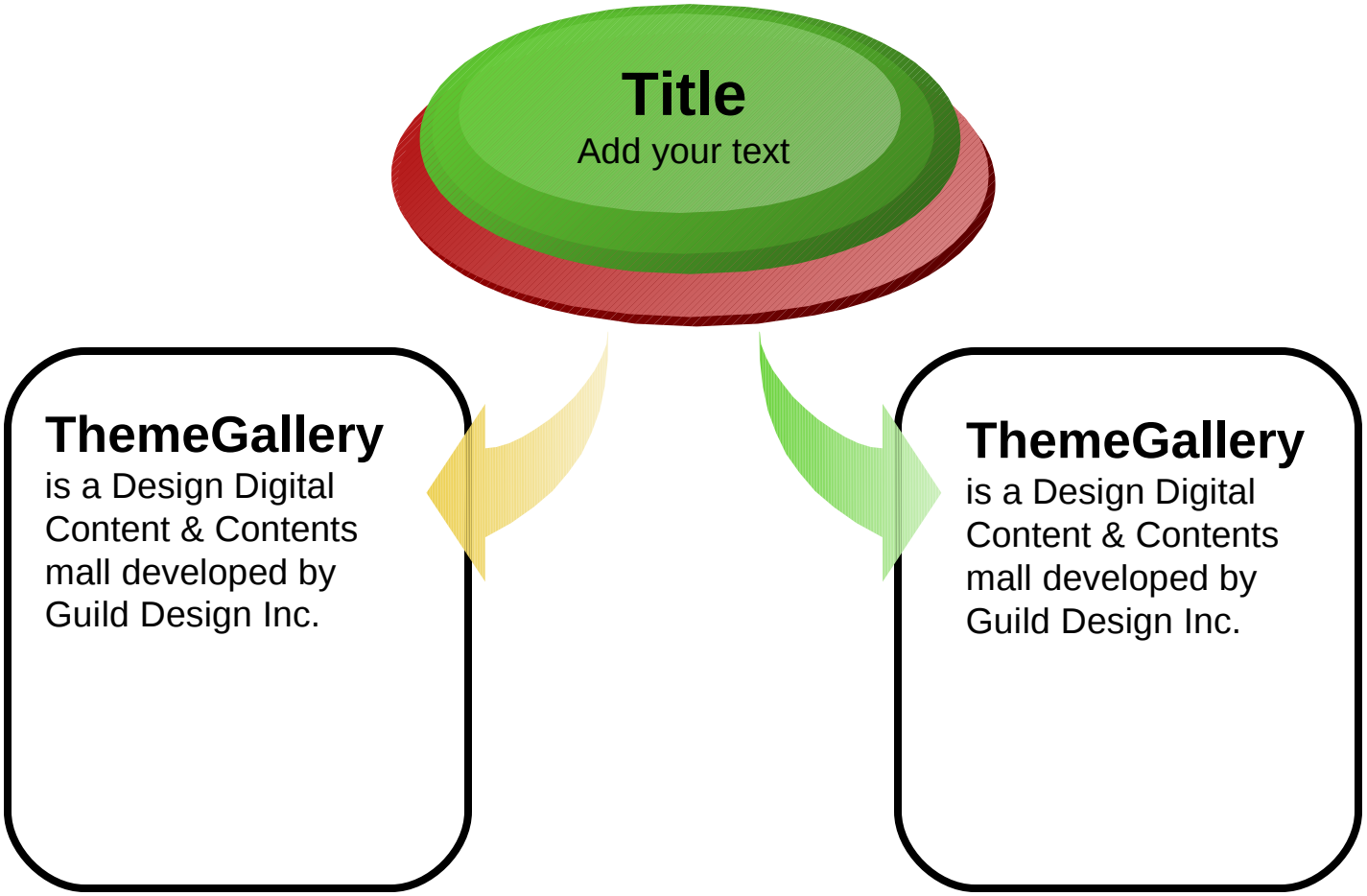
- Timer của hệ điều hành **Windows** là một dịch vụ định kỳ thông báo cho các ứng dụng biết rằng có một khoảng thời gian đã trôi qua.
- Thực hiện bằng cách sau một chu kỳ thời gian **Windows** gởi đến hàm window thông điệp **WM_TIMER**.

BỘ ĐỊNH THỜI TIMER

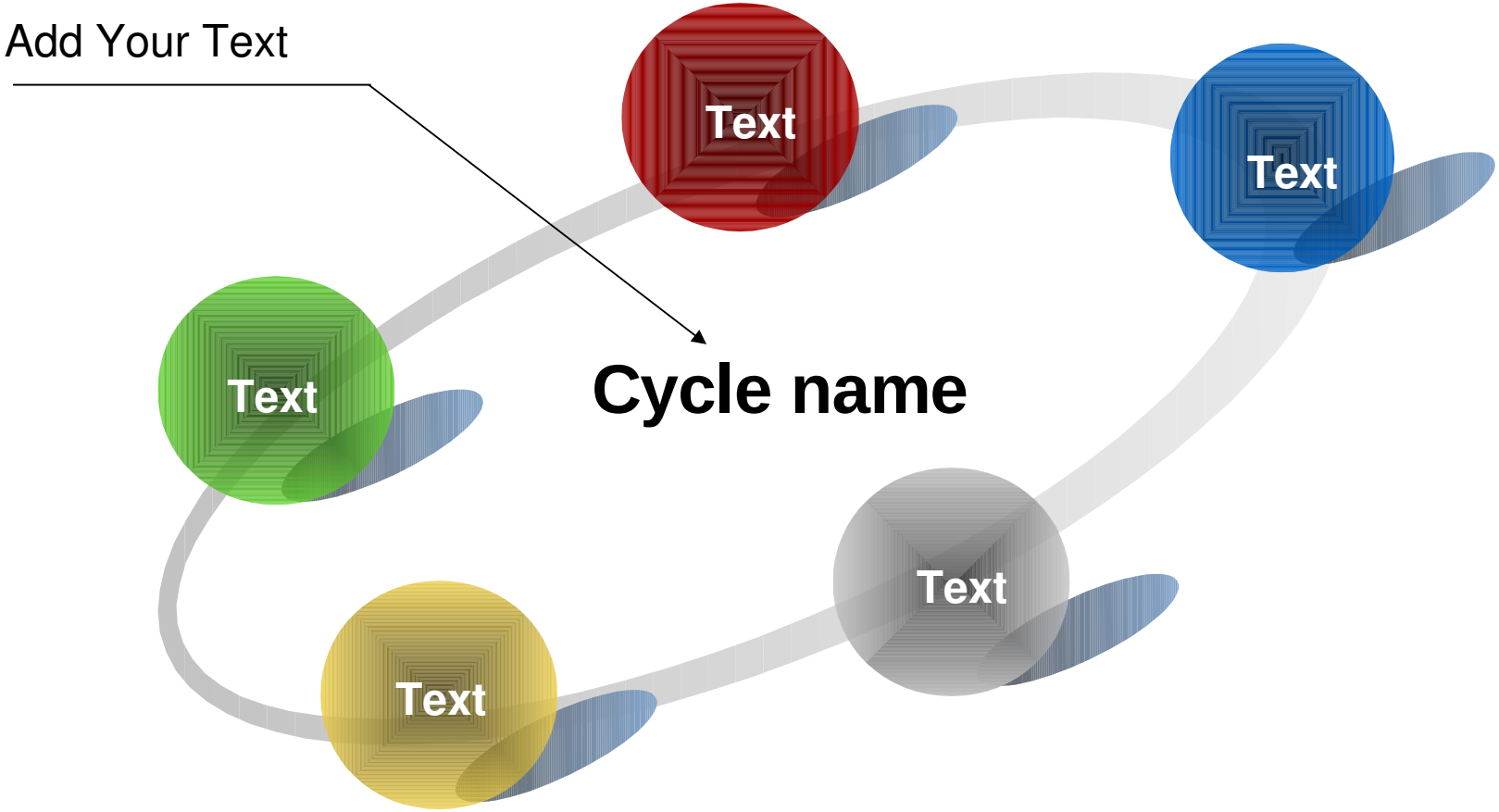
- Thiết lập bộ định thời: **sử dụng hàm** SetTimer**.**
  - ***UINT SetTimer(UINT nIDEvent, UINT uElapse, TIMERPROC lpTimerFunc);***

    - **nIDEvent : Chỉ danh của bộ Timer**
    - **uElapse: Chu kỳ thời gian mà Windows gởi thông điệp WM_TIMER tới hàm windows (tính bằng mili giây)**
    - **lpTimerFunc: Địa chỉ hàm timer – Có thể là NULL**

Diagram


Title
Add your text
ThemeGallery
is a Design Digital Content & Contents mall developed by Guild Design Inc.
ThemeGallery
is a Design Digital Content & Contents mall developed by Guild Design Inc.

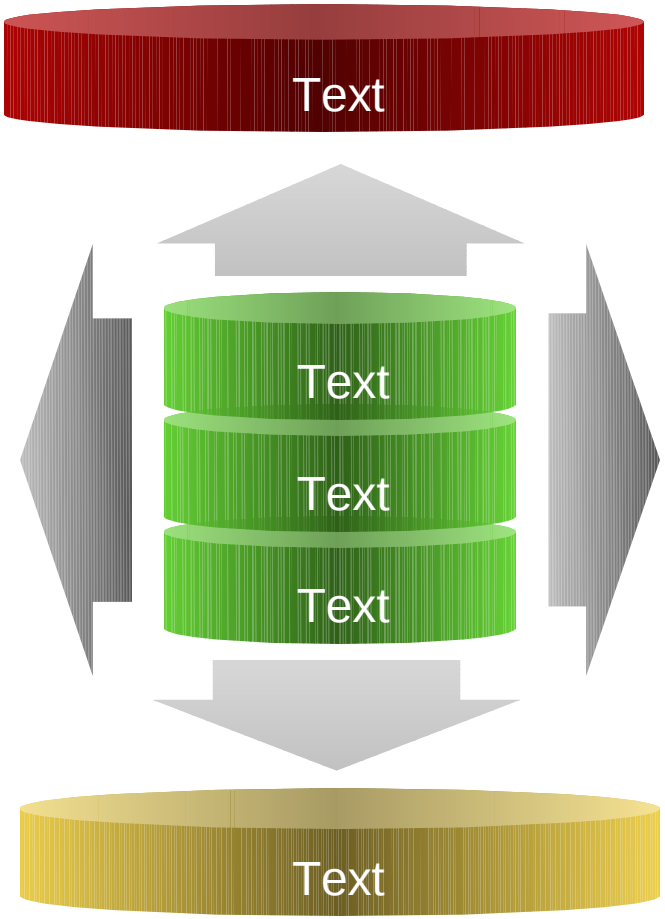
Cycle Diagram



Diagram

Add Your Title Text

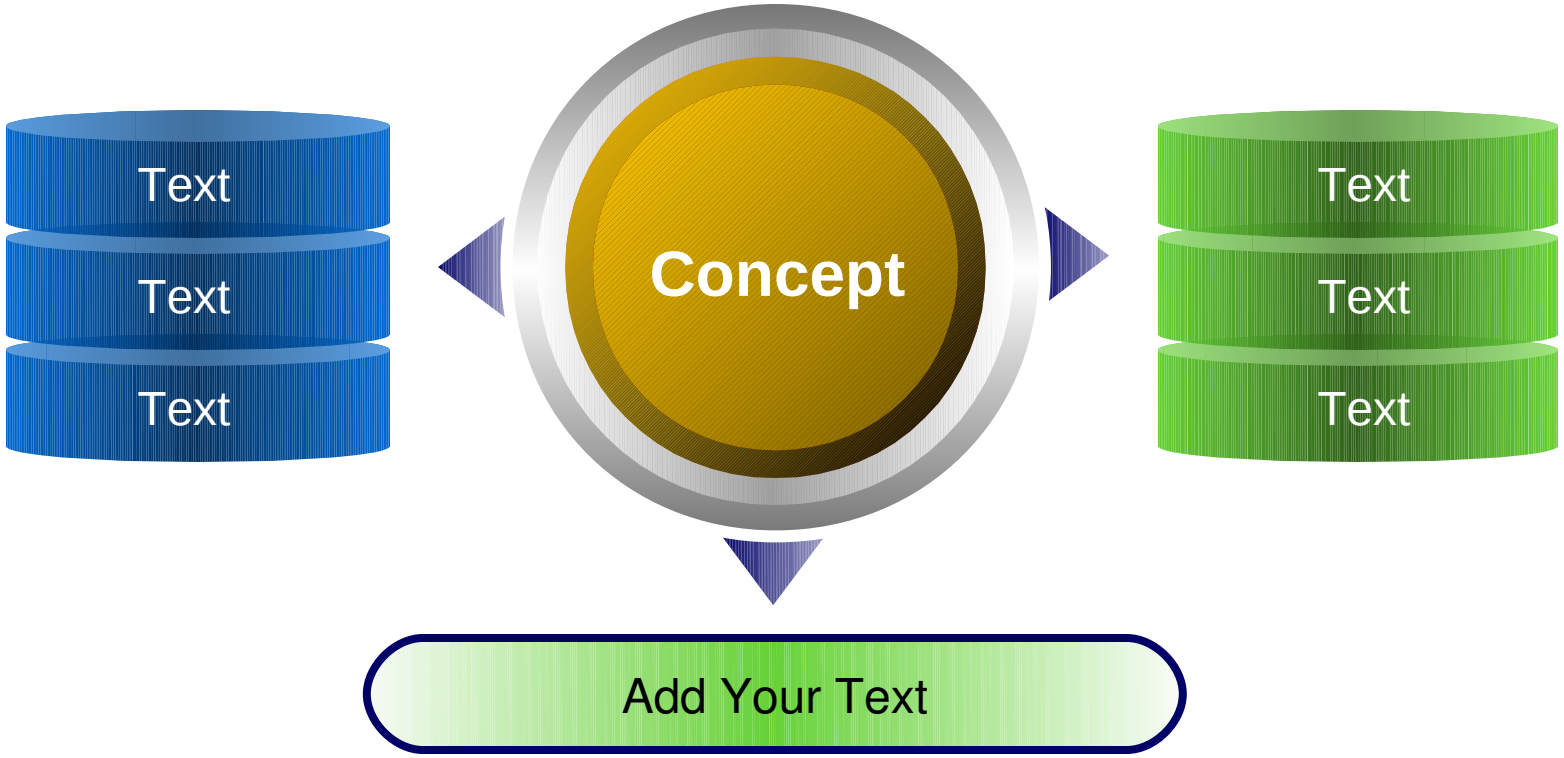
- Text 1
- Text 2
- Text 3
- Text 4
- Text 5



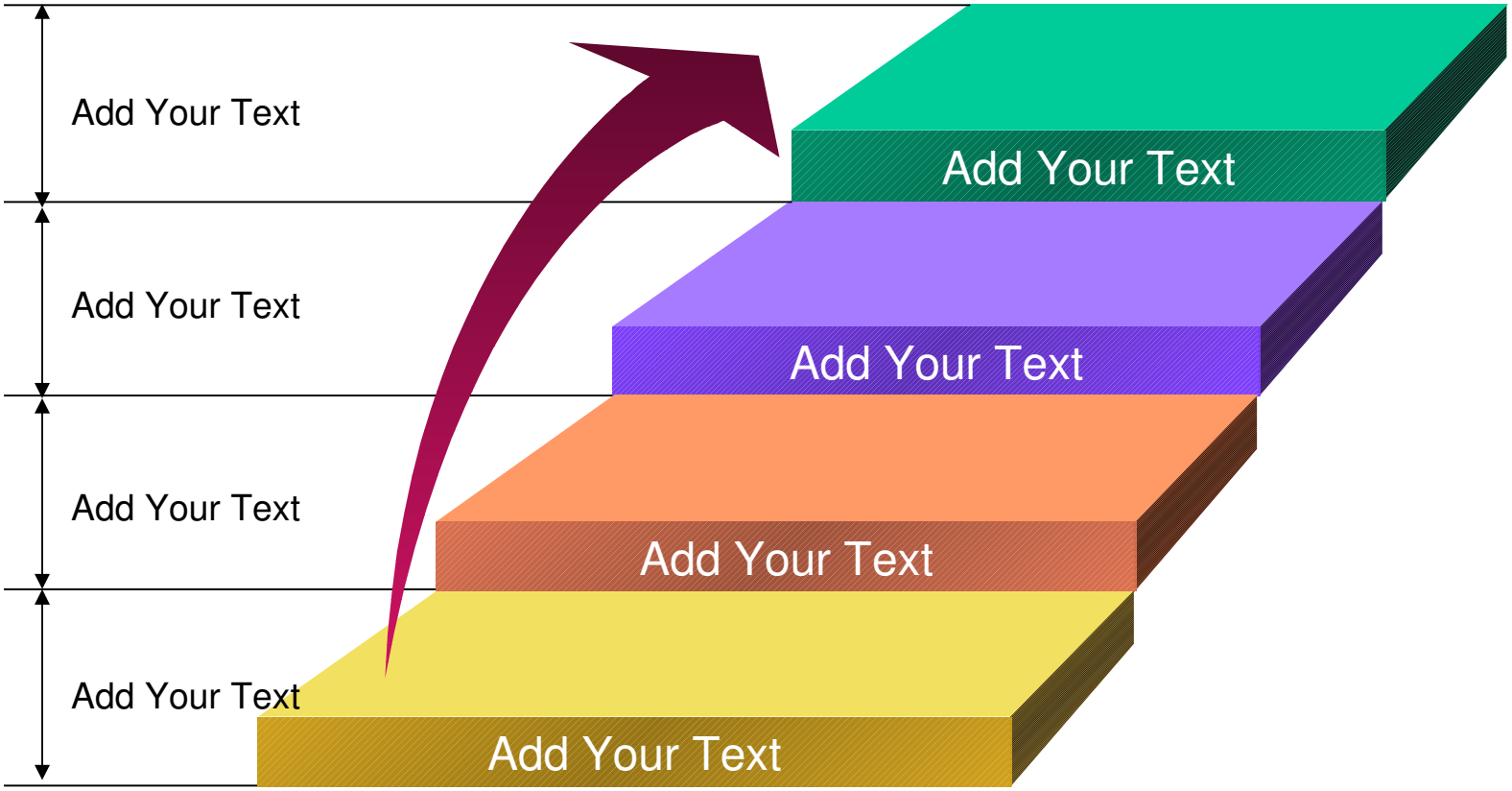
Add Your Title Text

- Text 1
- Text 2
- Text 3
- Text 4
- Text 5

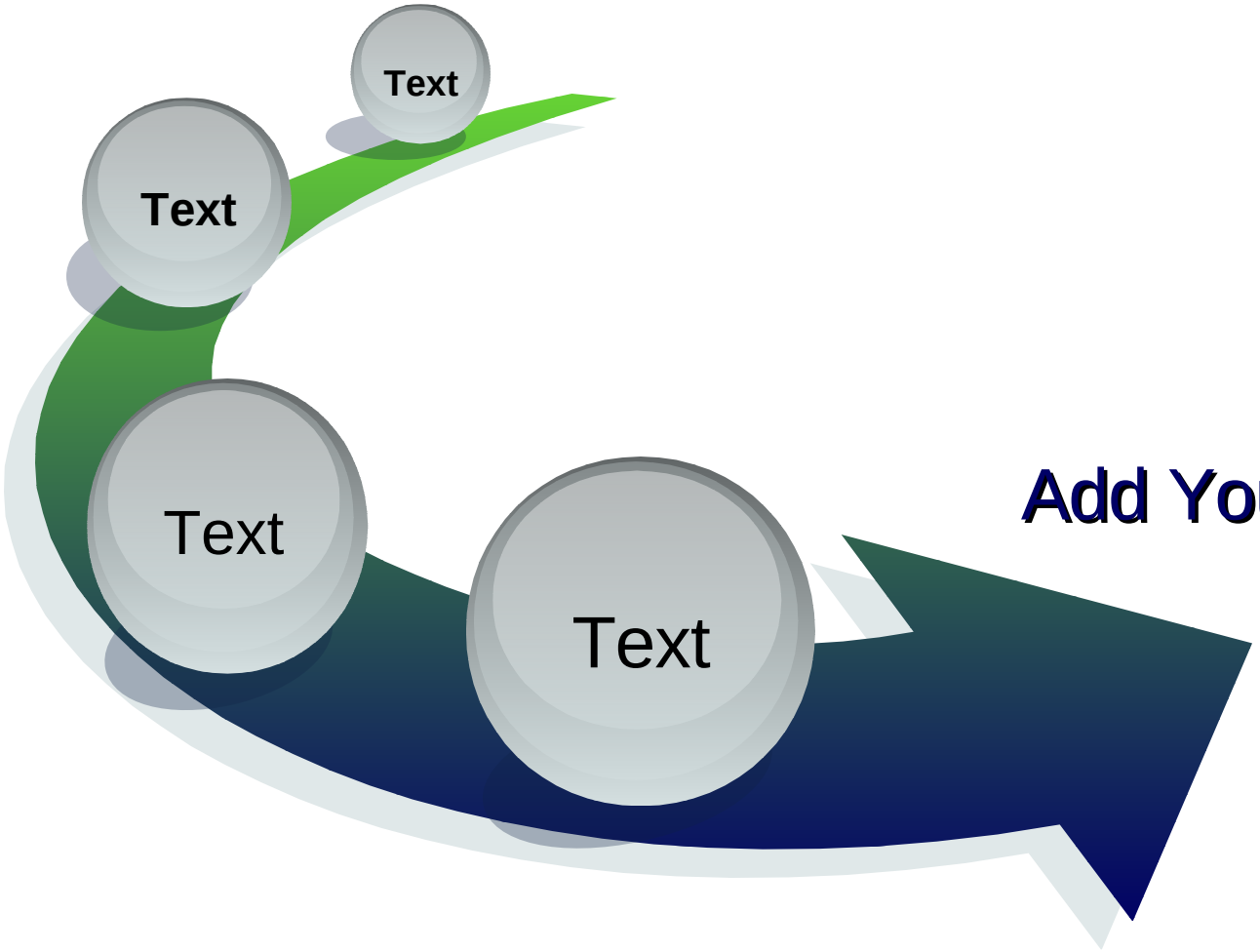
Diagram



Diagram

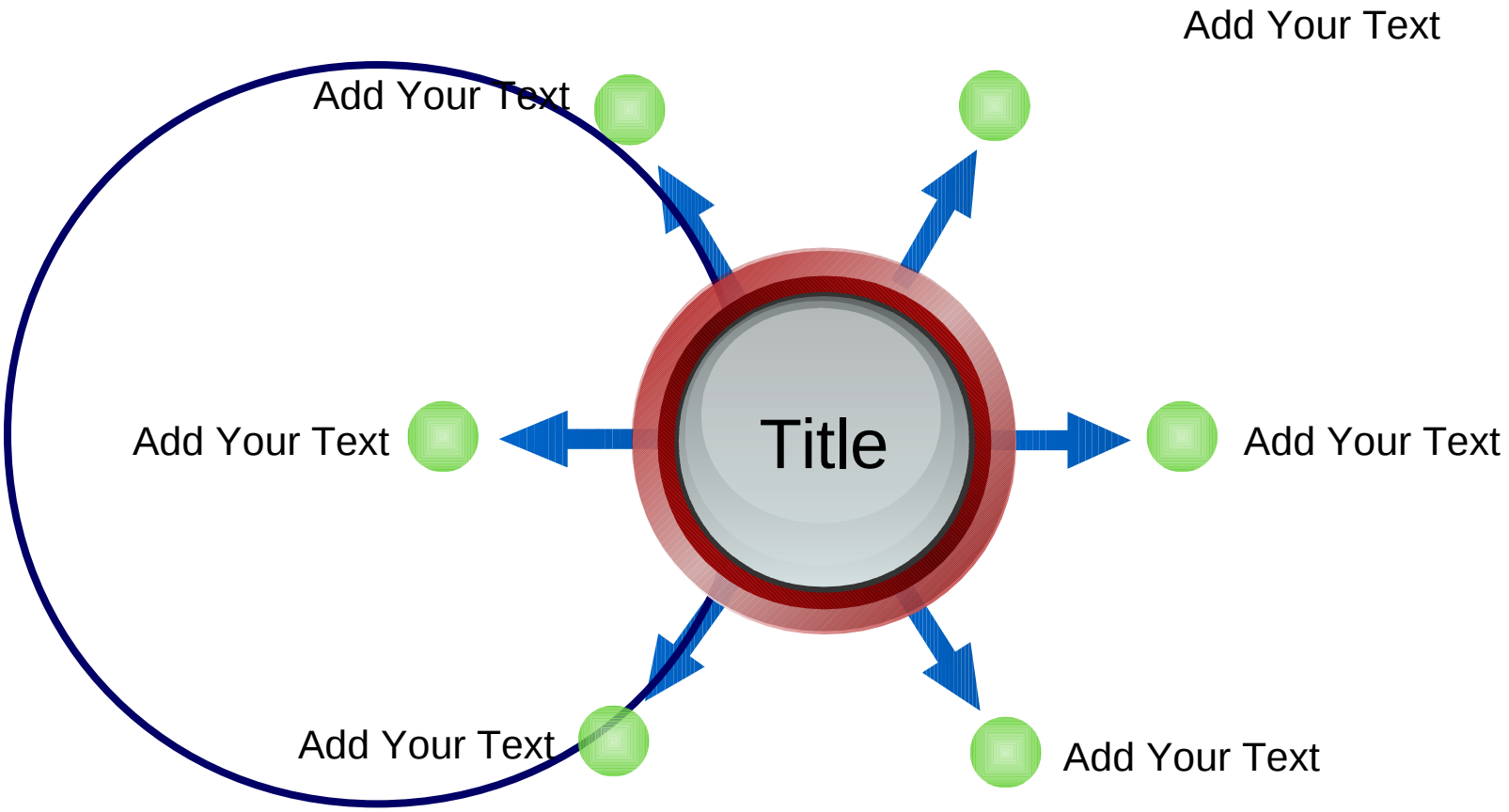

Add Your Text
Add Your Text
Add Your Text
Add Your Text
Add Your Text
Add Your Text
Add Your Text
Add Your Text

Diagram



Add Your Title

Diagram


Add Your Text
Add Your Text
Add Your Text
Title
Add Your Text
Add Your Text
Add Your Text

Diagram

1

ThemeGallery is a Design Digital Content & Contents mall developed by Guild Design Inc.

2

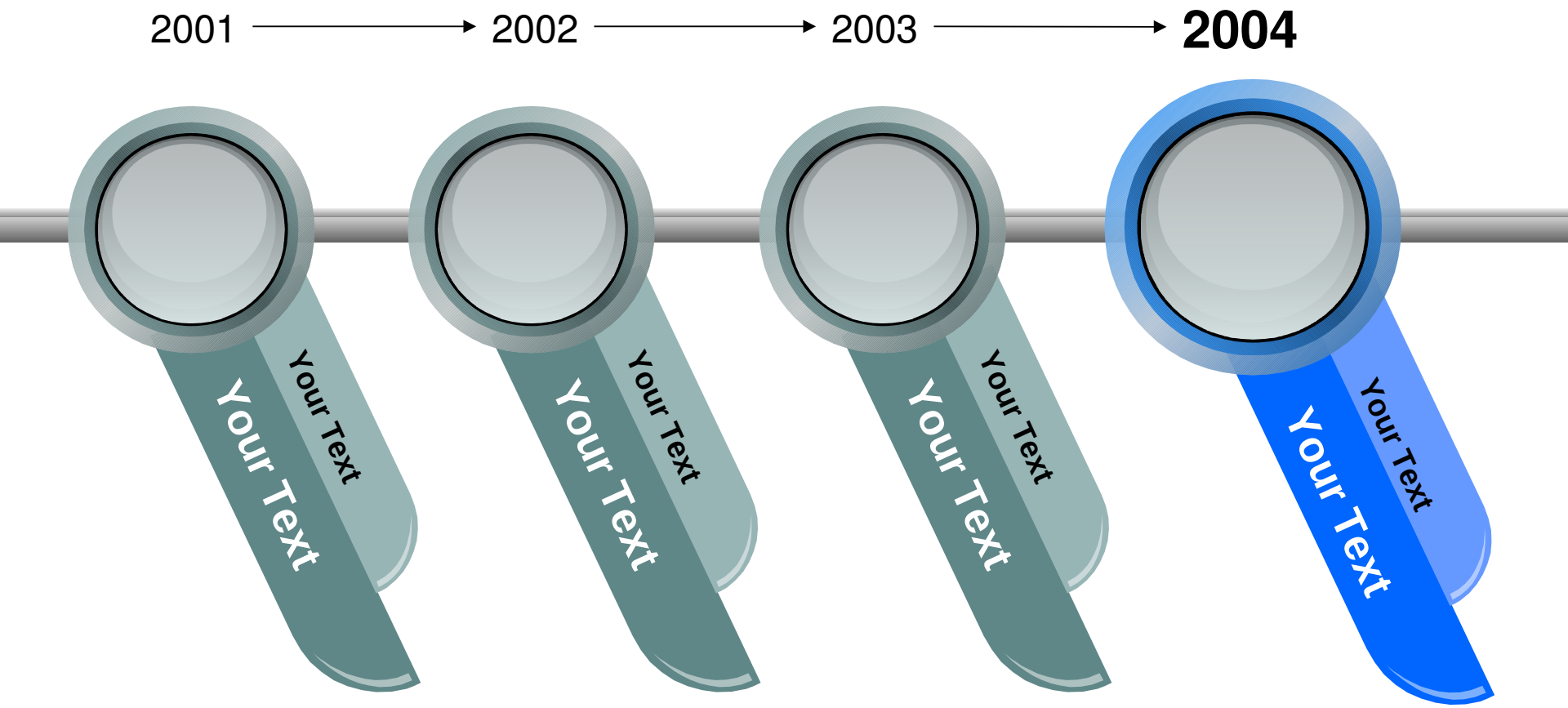
ThemeGallery is a Design Digital Content & Contents mall developed by Guild Design Inc.

3

ThemeGallery is a Design Digital Content & Contents mall developed by Guild Design Inc.

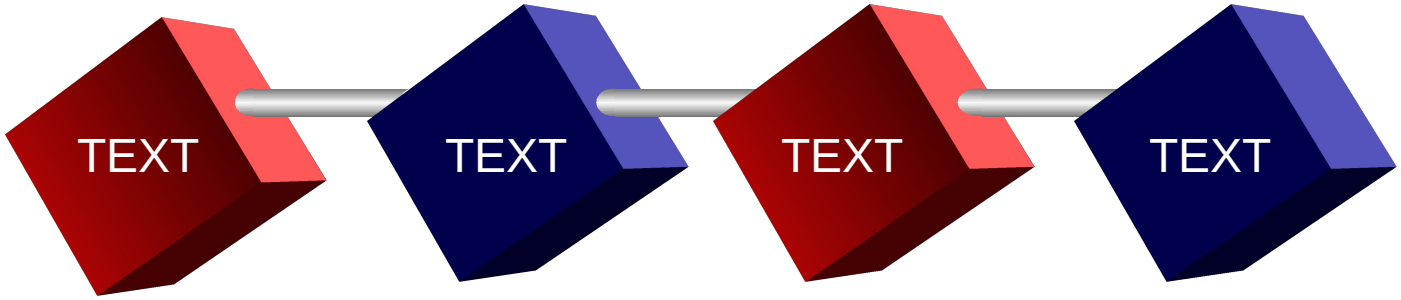
Diagram

Diagram


2001
2002
2003
2004
Your Text
Your Text
Your Text
Your Text
Your Text
Your Text
Your Text
Your Text

Progress Diagram

Block Diagram

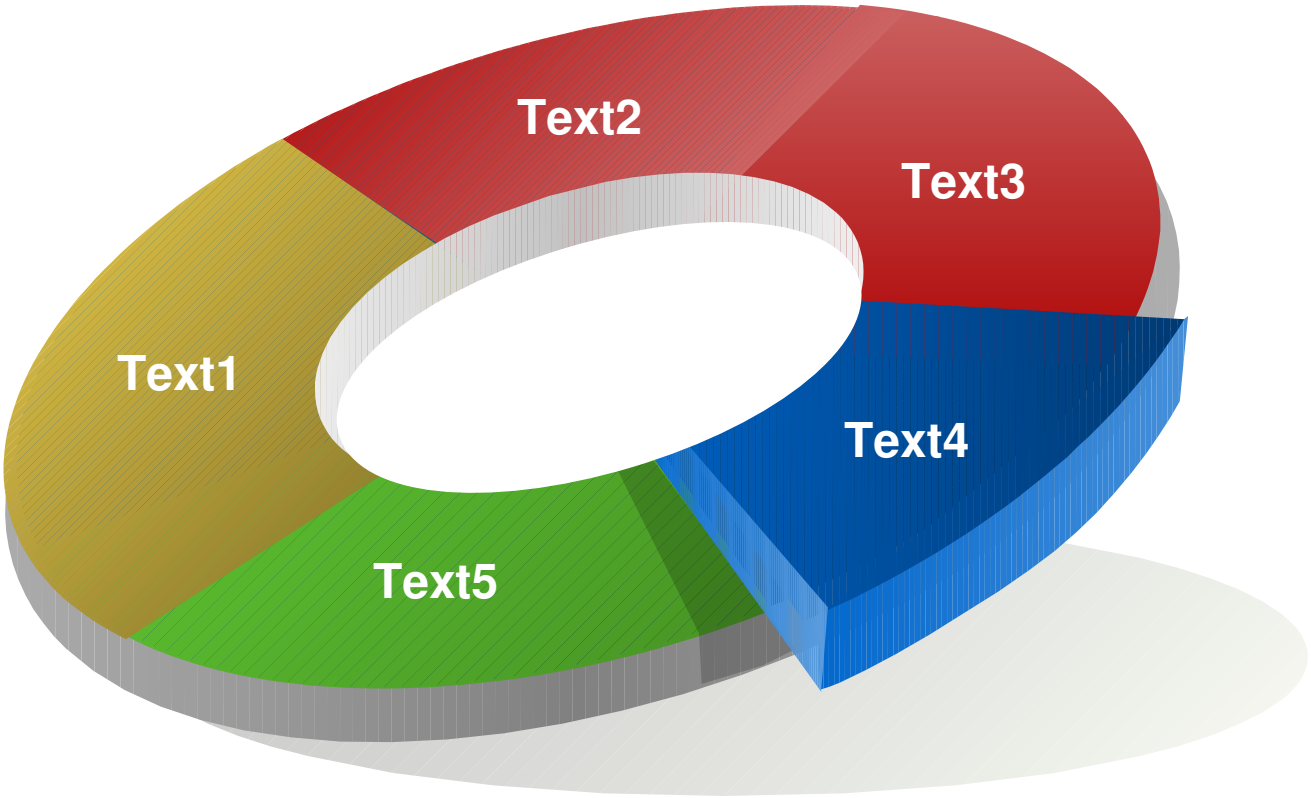


TEXT	TEXT	TEXT	TEXT

Table

	TEXT	TEXT	TEXT	TEXT	TEXT
Title A					
Title B					
Title C					
Title D					
Title E					
Title F					

3-D Pie Chart



Marketing Diagram

Add Your Text

Thank You !

www.themegallery.com